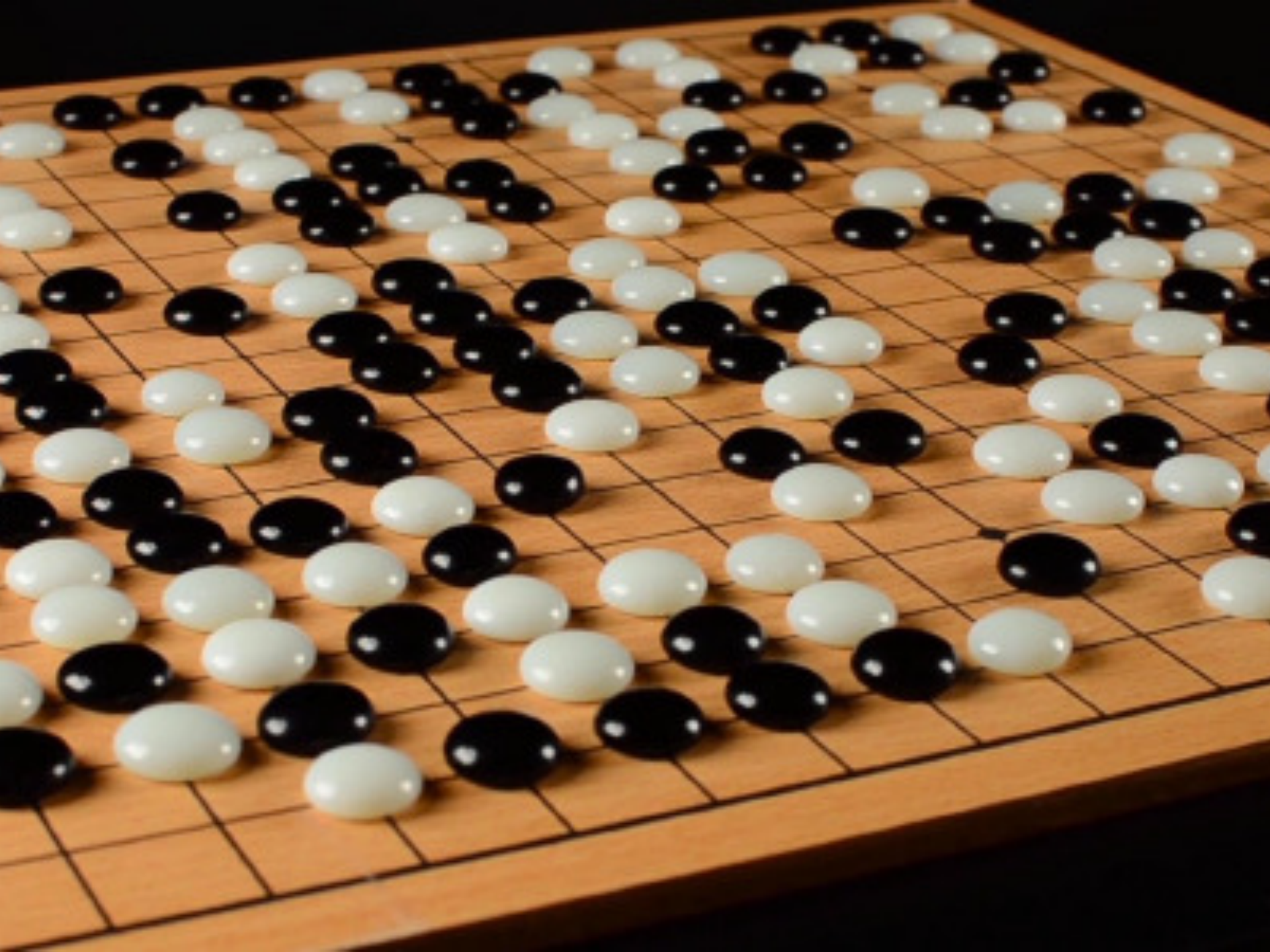
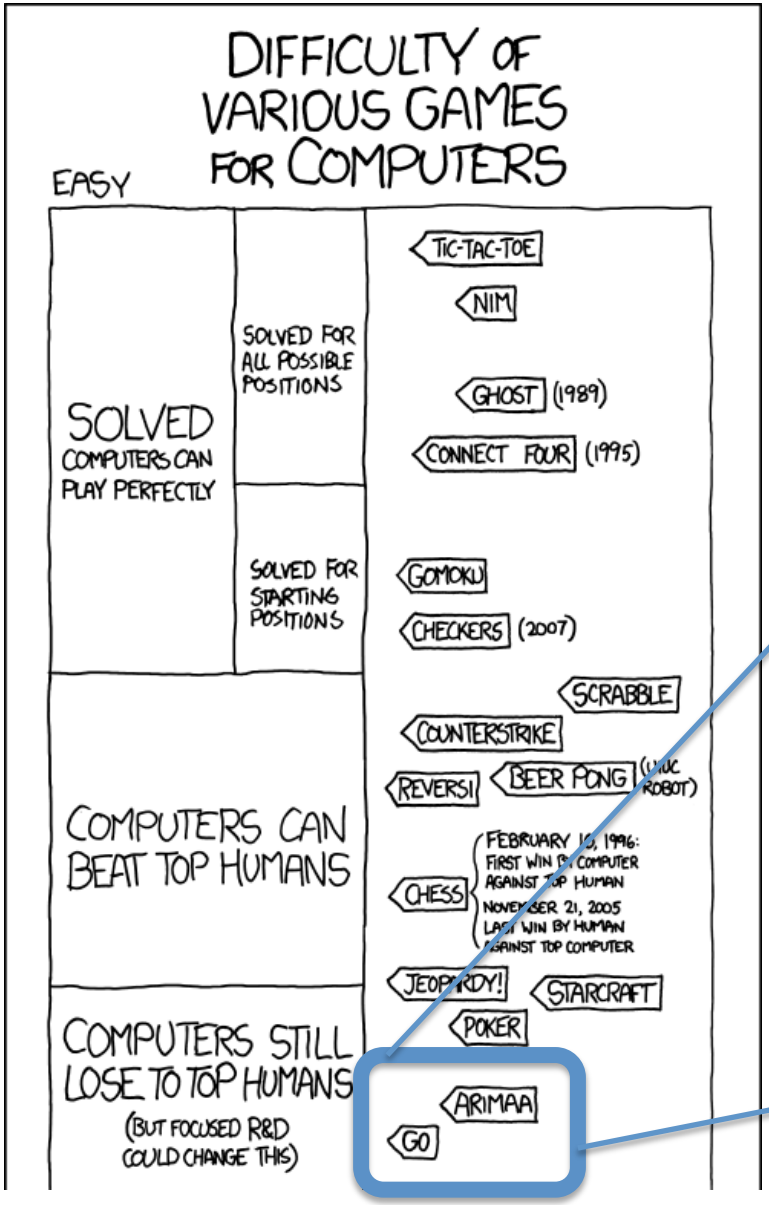
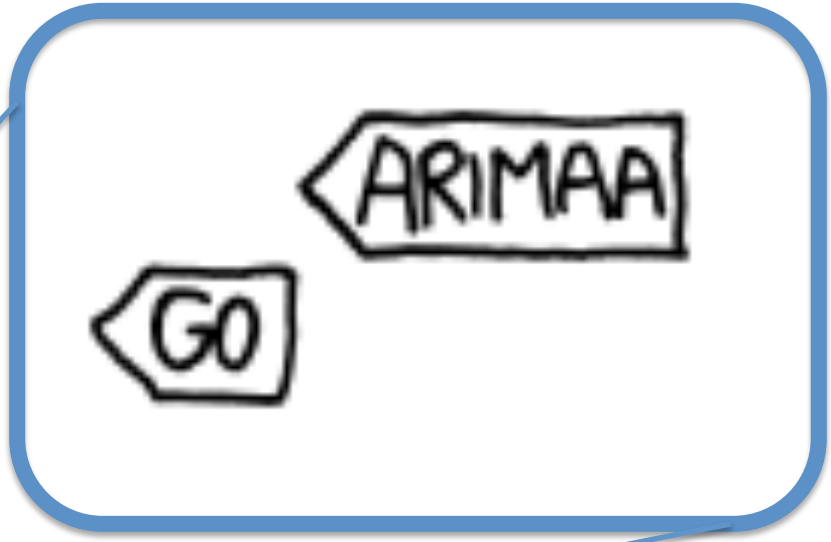
Computer Science has Changed

Hit a huge milestone…



The Hardest Game Left





How they beat humans is meaningful

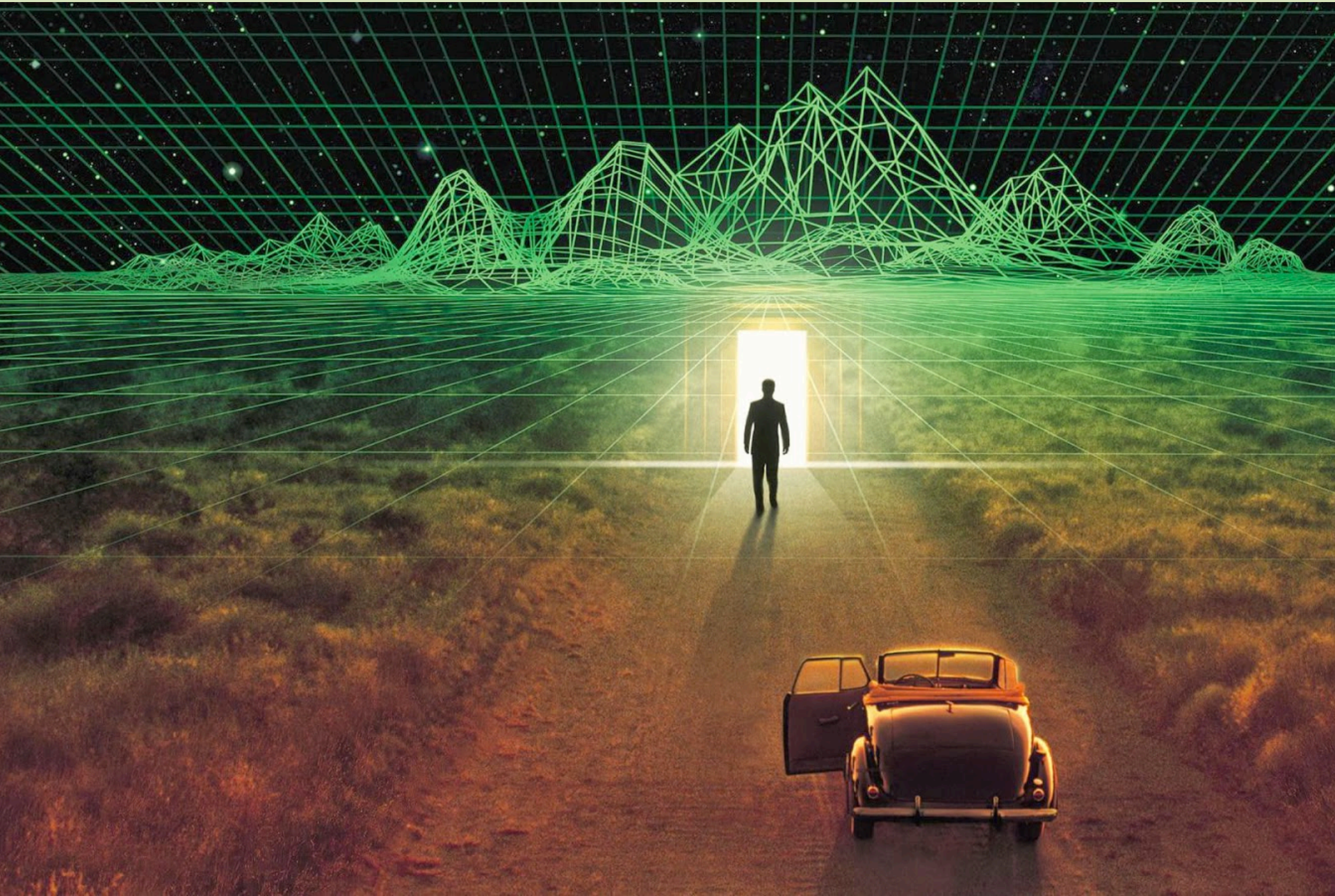
Big O and Games

Chris Piech

CS 106B
Lecture 10
Jan 29, 2015

Partners After Class

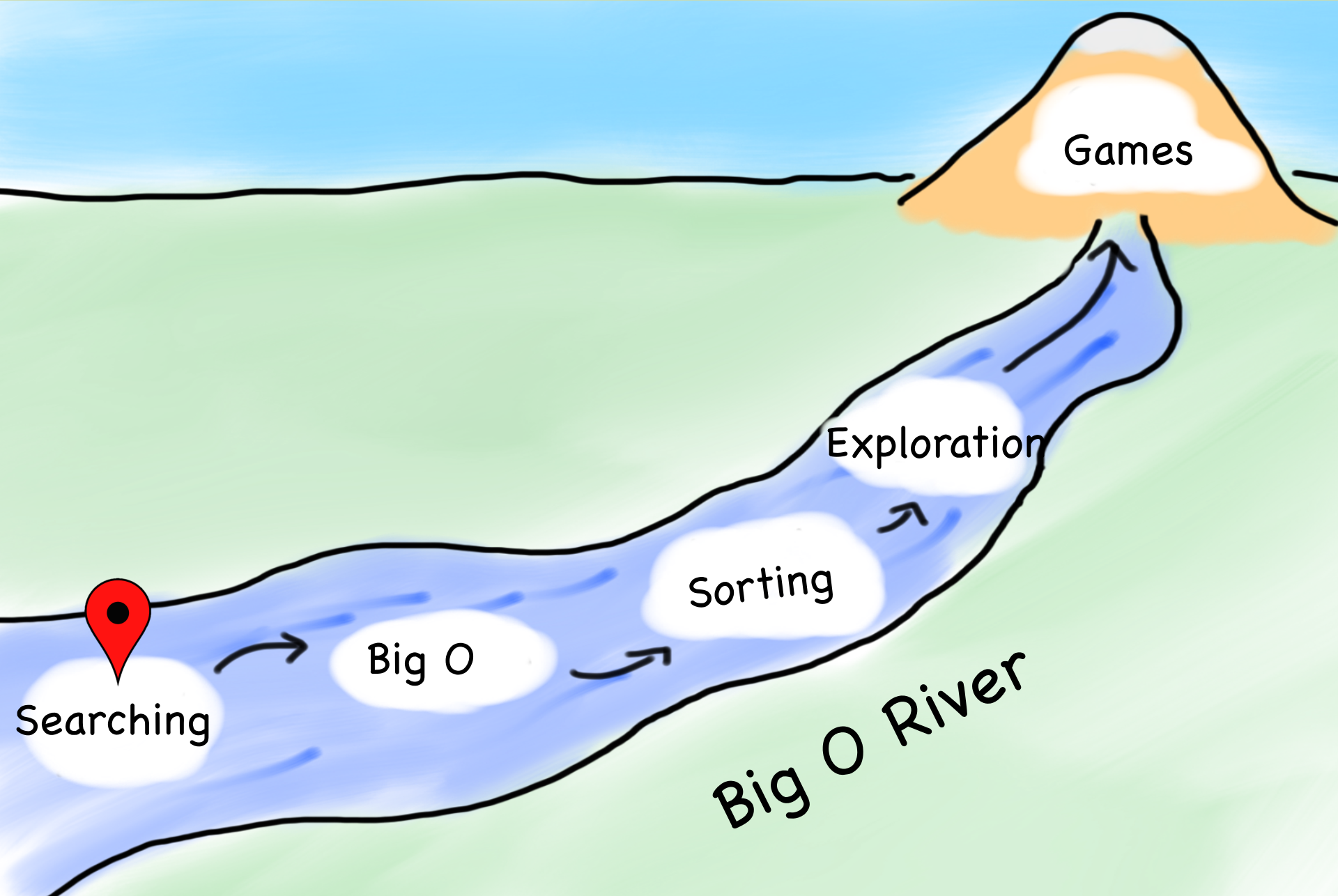
Contest Due Today



Today's Goals

1. Understand Big O
2. Get a feel for exponential growth

Today's Goals



Searching

Linear Search

201	202	203	205	206	207	208	209	210	212	213	214	215	216	217	218	219	224	225	228	229	231
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
234	239	240	248	251	252	253	254	256	260	262	267	269	270	276	281	283	301	302	303	304	305
22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
307	308	309	310	312	313	314	315	316	317	318	319	320	321	323	325	330	331	334	336	337	339
44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65
347	351	352	360	361	364	385	386	401	402	404	405	406	407	408	409	410	412	413	414	415	416
66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87
417	419	423	424	425	430	432	434	435	440	443	445	469	470	475	478	479	480	484	501	502	503
88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109
504	505	507	508	509	510	512	513	515	516	517	518	520	530	540	541	551	559	561	562	563	564
110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131
567	570	571	573	574	575	580	585	586	601	602	603	605	606	607	608	609	610	612	614	615	616
132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153
617	618	619	620	623	626	630	631	636	641	646	650	651	660	661	662	678	682	701	702	703	704
154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
706	707	708	712	713	714	715	716	717	718	719	720	724	727	731	732	734	740	754	757	760	762
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197
763	765	769	770	772	773	774	775	779	781	785	786	801	802	803	804	805	806	808	810	812	813
198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219
814	815	816	817	818	828	830	831	832	835	843	845	847	848	850	856	857	858	859	860	862	863
220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241
864	865	870	878	901	903	904	906	907	908	909	910	912	913	914	915	916	917	918	919	920	925
242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263
928	931	936	937	940	941	947	949	951	952	954	956	959	970	971	972	973	978	979	980	985	989
264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285

Mordor

One does not simply walk into it...

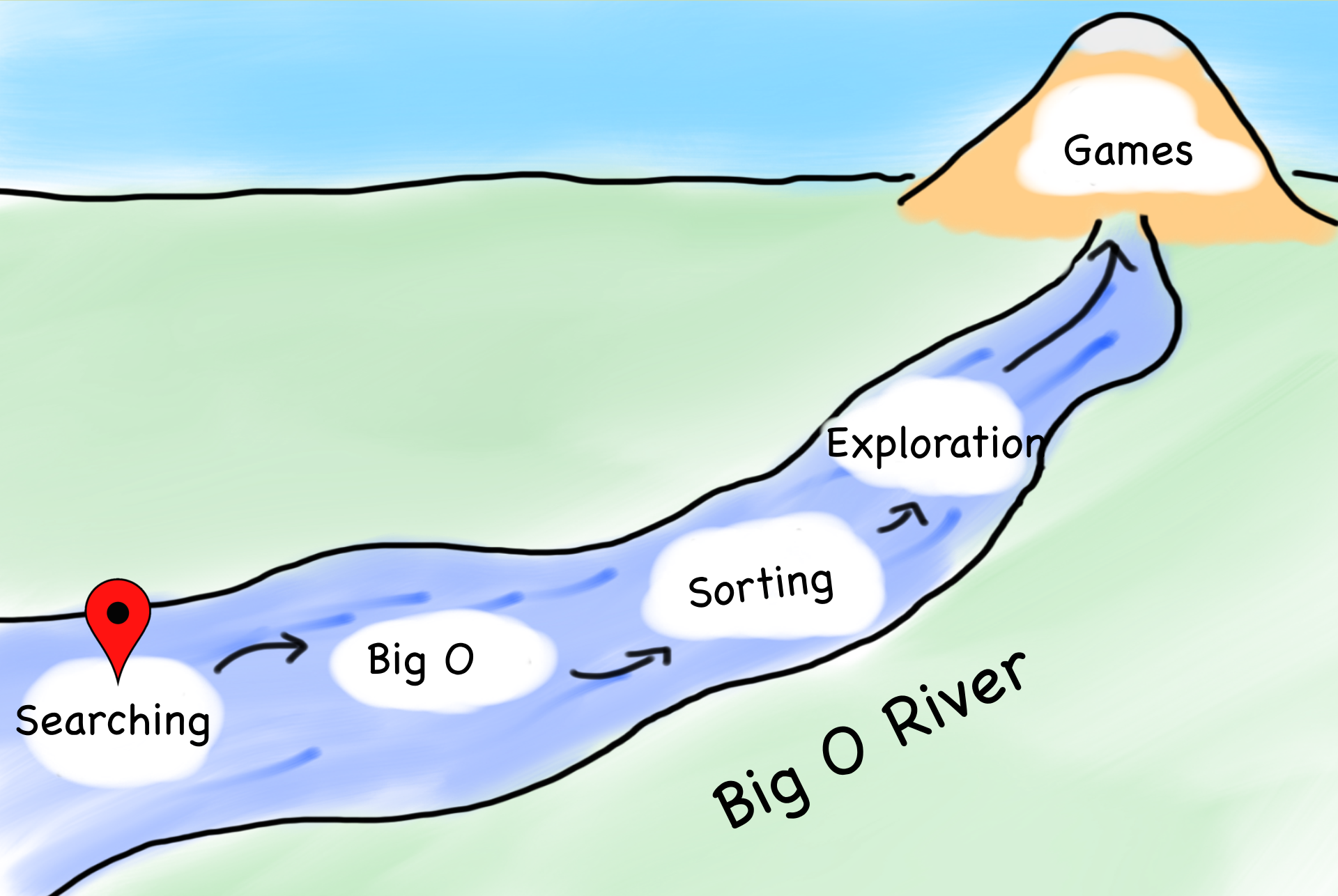
Binary Search

Binary search needs to look at only eight elements to find 650.

201	202	203	205	206	207	208	209	210	212	213	214	215	216	217	218	219	224	225	228	229	231
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
234	239	240	248	251	252	253	254	256	260	262	267	269	270	276	281	283	301	302	303	304	305
22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
307	308	309	310	312	313	314	315	316	317	318	319	320	321	323	325	330	331	334	336	337	339
44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65
347	351	352	360	361	364	385	386	401	402	404	405	406	407	408	409	410	412	413	414	415	416
66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87
417	419	423	424	425	430	432	434	435	440	443	445	469	470	475	478	479	480	484	501	502	503
88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109
504	505	507	508	509	510	512	513	515	516	517	518	520	530	540	541	551	559	561	562	563	564
110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131
567	570	571	573	574	575	580	585	586	601	602	603	605	606	607	608	609	610	612	614	615	616
132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153
617	618	619	620	623	626	630	631	636	641	646	650	651	660	661	662	678	682	701	702	703	704
154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
706	707	708	712	713	714	715	716	717	718	719	720	724	727	731	732	734	740	754	757	760	762
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197
763	765	769	770	772	773	774	775	779	781	785	786	801	802	803	804	805	806	808	810	812	813
198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219
814	815	816	817	818	828	830	831	832	835	843	845	847	848	850	856	857	858	859	860	862	863
220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241
864	865	870	878	901	903	904	906	907	908	909	910	912	913	914	915	916	917	918	919	920	925
242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263
928	931	936	937	940	941	947	949	951	952	954	956	959	970	971	972	973	978	979	980	985	989
264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285

How Can We Compare

Today's Goals



Today's Goals

Timing?

In C++, you can measure elapsed time by calling the `time` function

1. The `time` function is often too rough.

2. Some measurements are likely to be wildly off.

3. Different computers.

4. Different translation of code.

What can we do?

suspense

Count number of executions.

Count Number of Executions

Think about the *worst* case.

```
int find(Vector<int> & vec, int goal) {
  for(int i = 0; i < vec.size(); i++) {
    if(vec[i] == goal) return i;
  }
  return -1;
}
```

$2n + 1$

$2n$

1

$$T(n) = 3n + 2$$

Do we really care about the 3?

Do we really care about the +2?

Big O

We say a function $f(n)$ is **"big-O"** of another function $g(n)$, and write "$f(n)$ is $\boldsymbol{O}(g(n))$" if there exist positive constants c and n_0 such that:

$$f(n) \leq c \cdot g(n)$$

For all

$$n > n_0$$

Count Number of Executions

$T(n) = 3n + 2$ Where $n_0 = 1$

$T(n) \leq 3n + 2n$

$T(n) \leq 5n$

$T(n)$ is $O(n)$ Where $c = 5$

We say that $f(n)$ Is big $O(g(n))$ if:

$$f(n) \leq c \cdot g(n)$$

For all

$$n > n_0$$

Big O of Binary Search?

Comparing Big O

Number of steps required for linear vs. binary search:

linear N	binary log N
10	3
100	7
1000	10
1,000,000	20
1,000,000,000,000,000	60

Today's Goals

Today's Goals

Sorting

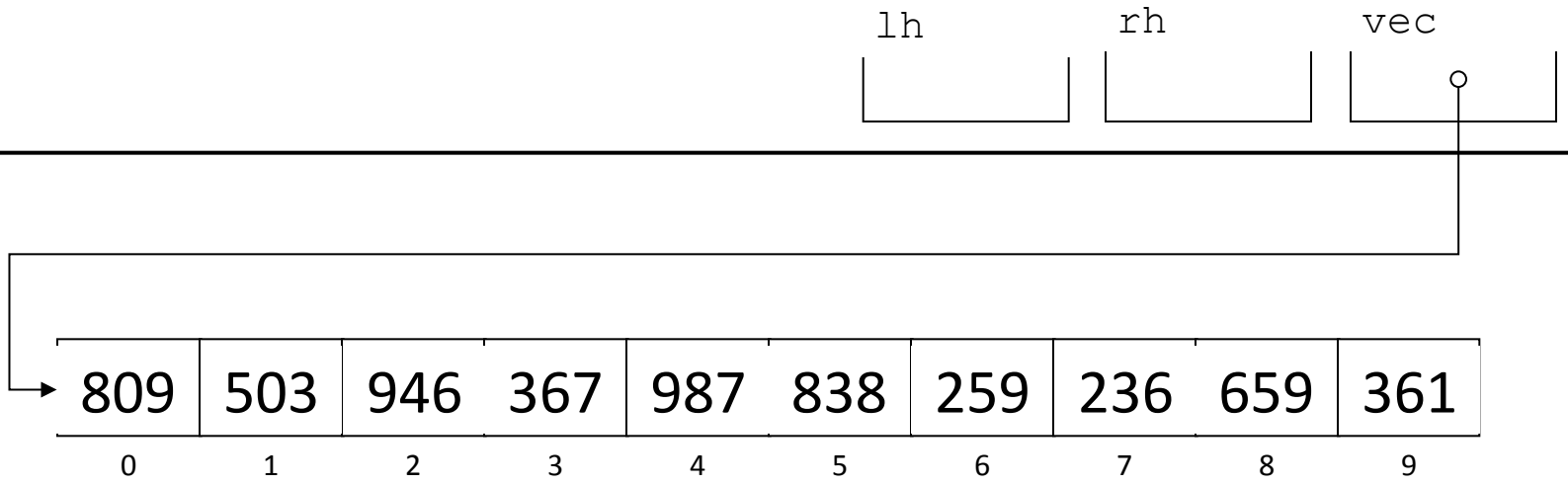
Even Presidents Have to Know

http://www.youtube.com/watch?v=k4RRi_ntQc8

Selection Sort

```
int main() {
```

```
 void Sort(Vector<int> & vec) {
    for ( int lh = 0 ; lh < vec.size() ; lh++ ) {
       int rh = FindSmallest(vec, lh, vec.size() - 1);
       Swap(vec[lh], vec[rh]);
    }
 }
```



skip simulation

Gauss Summation

Gauss Summation

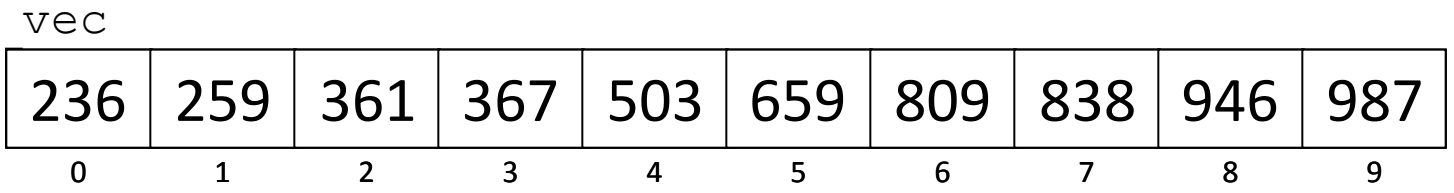
Selection Sort Complexity

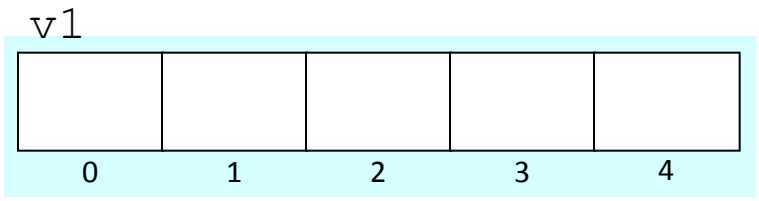
$$\mathcal{O}(n^2)$$

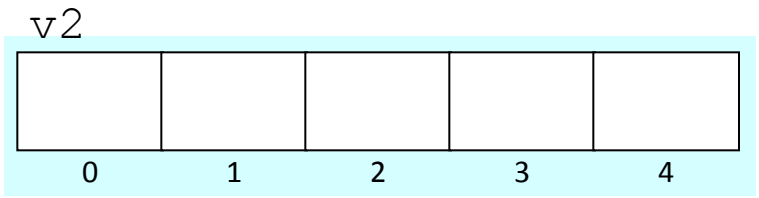
But wait! Check this out…

Merge Sort

1. Divide the vector into two halves: `v1` and `v2`.
2. Sort each of `v1` and `v2` recursively.
3. Clear the original vector.
4 Merge elements into the original vector by choosing the smallest element from `v1` or `v2` on each cycle.







Merge Sort Complexity

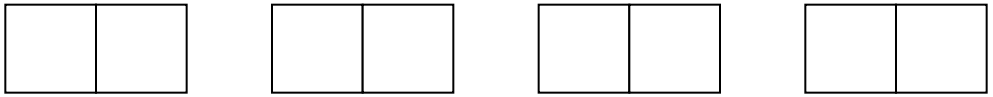
Sorting 8 items

requires

Two sorts of 4 items

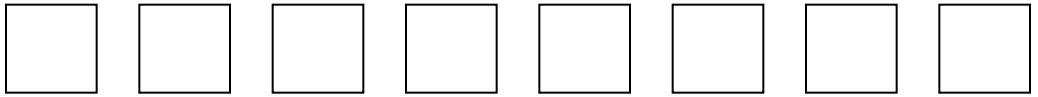
which requires

Four sorts of 2 items



which requires

Eight sorts of 1 item



The work done at each level (*i.e.*, the sum of the work done by all the calls at that level) is proportional to the size of the level. Each of which are *N*. The running time is therefore proportional to *N* times the number of levels.

Merge Sort Complexity

$$\mathcal{O}(n \log n)$$

Comparing Sort

The difference between $O(N^2)$ and $O(N \log N)$ is enormous for large values of N, as shown in this table:

N	N^2	$N \log_2 N$
10	100	30
100	10,000	700
1,000	1,000,000	10,000
10,000	100,000,000	100,000
100,000	10,000,000,000	2,000,000
1,000,000	1,000,000,000,000	20,000,000

Socrative

a) O(1) b) O(logN) c) O(N) d) $O(N^2)$

```
for (int k = n; k >= 0; k--) {
   cout << k << endl;
}
```

$O(N)$

This loop executes n times.

```
for (int i = 0; i < n; i++) {
   for (int j = 0; j < i; j++) {
      cout << j << endl;
   }
}
```

$O(N^2)$

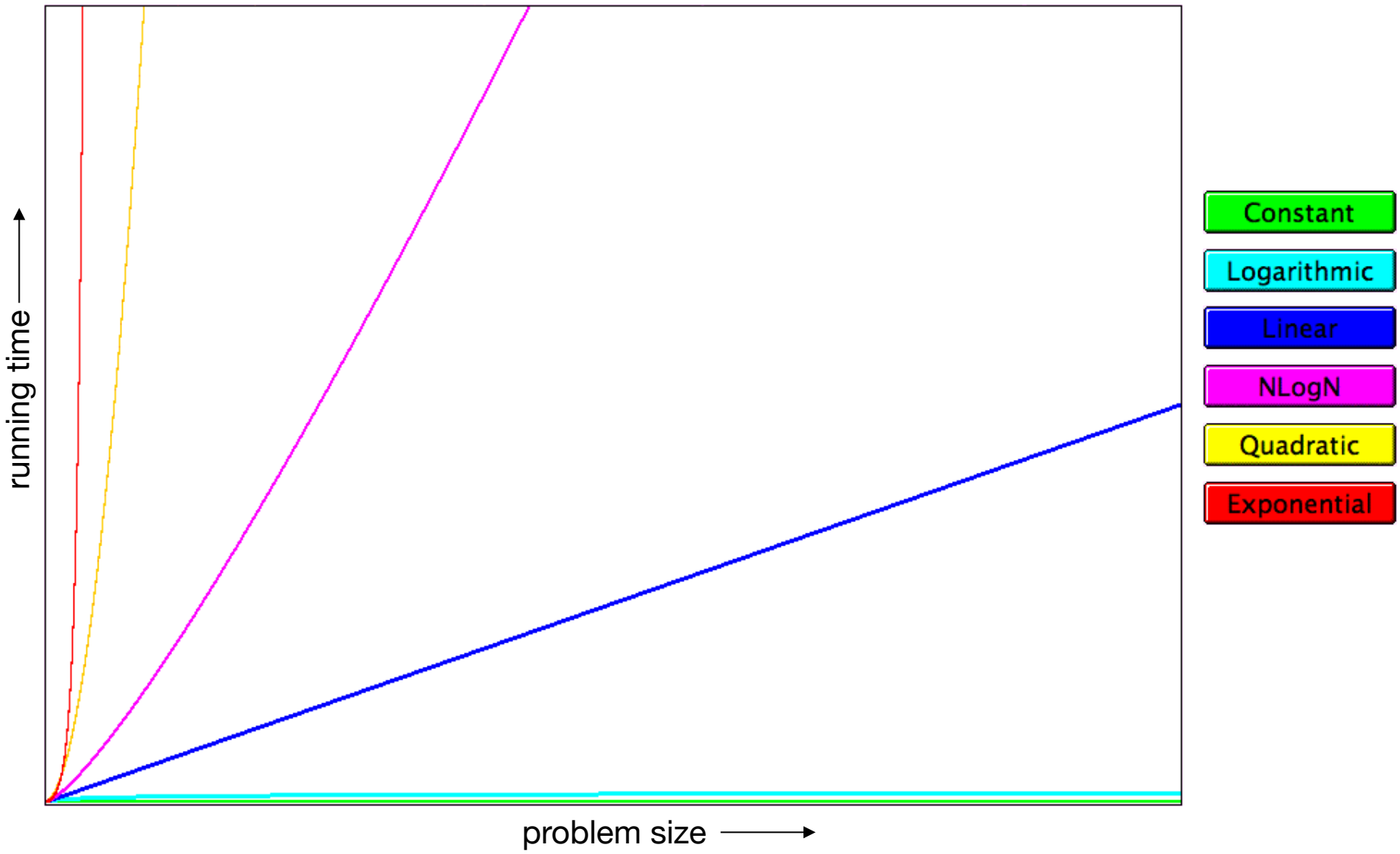
This loop follows the pattern from selection sort.

```
for (int i = 0; i < 100; i++) {
   cout << i << endl;
}
```

$O(1)$

This loop does not depend on the value of `n` *at all.*

Complexity Classes

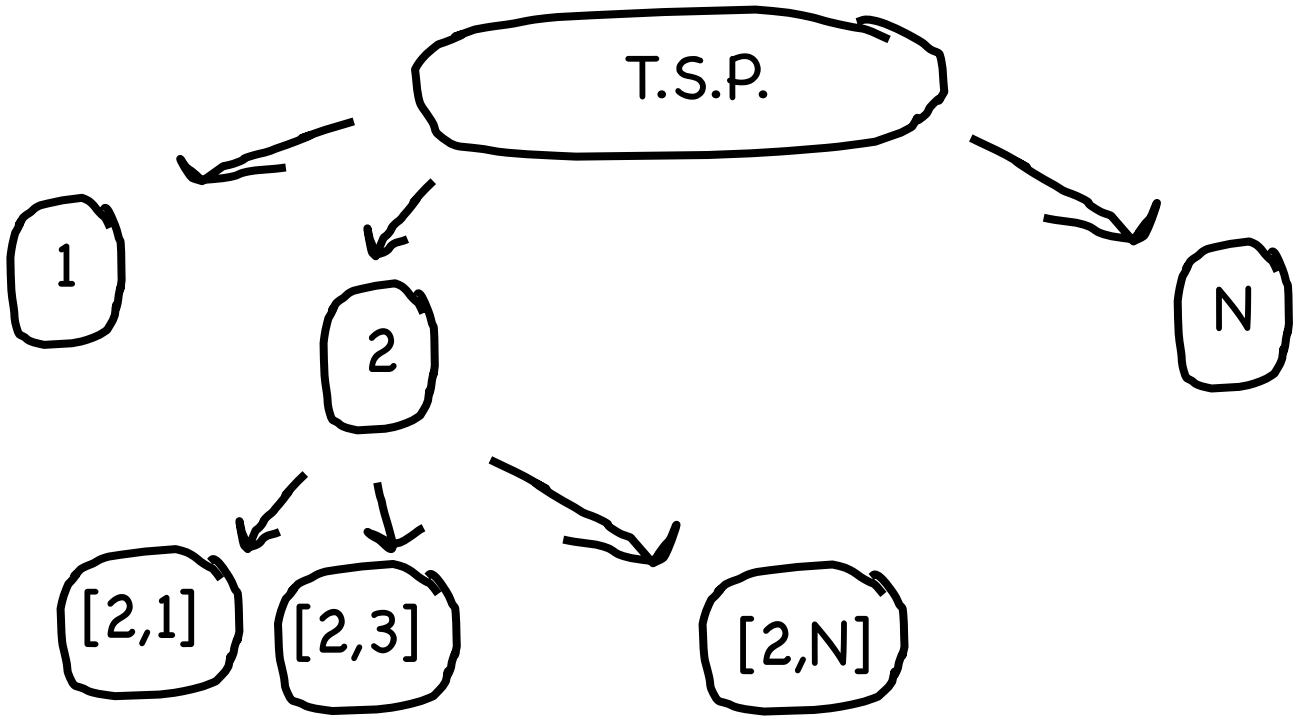


Today's Goals

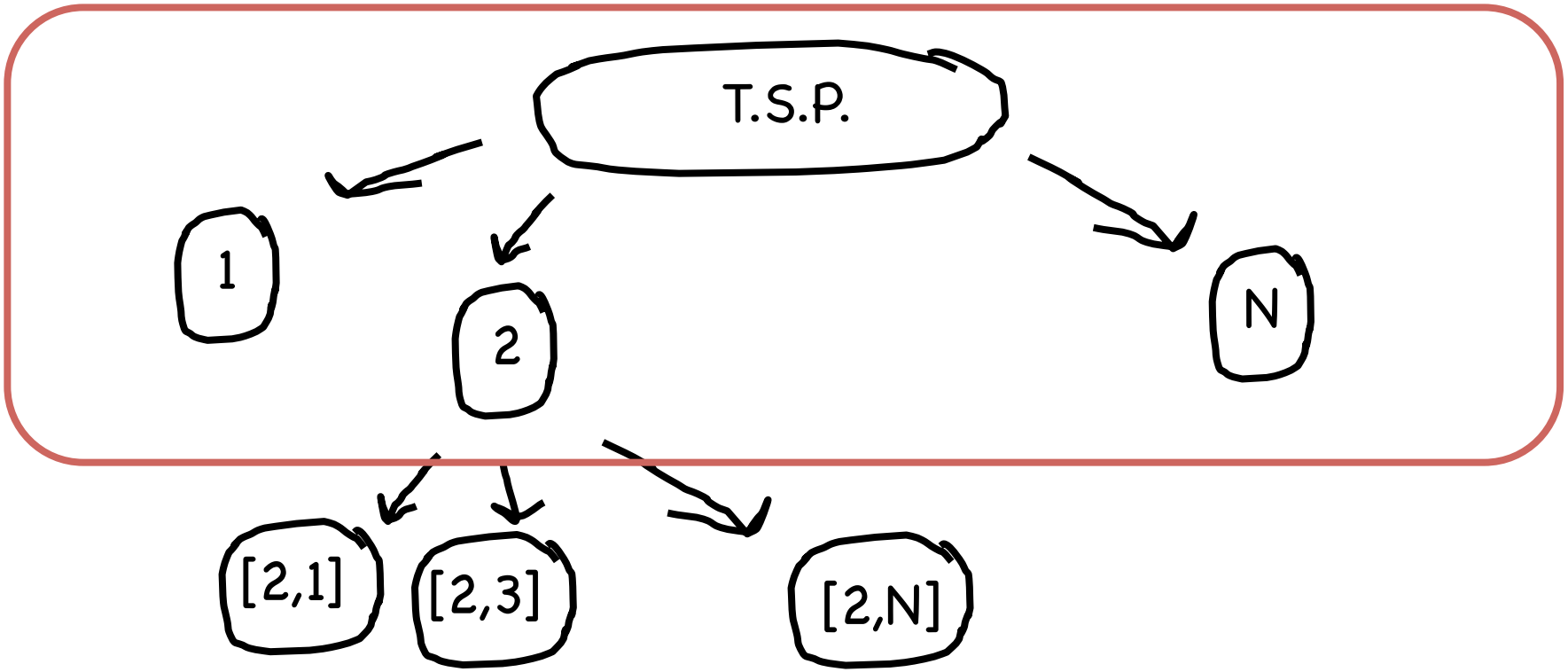
Today's Goals


nationalatlas.gov™
STATES AND CAPITALS
Traveling Salesperson Problem:
We have a bunch of cities to visit. In what order should we visit them to minimize total travel distance?
WASHINGTON
Olympia
OREGON
Salem
CALIFORNIA
Sacramento
NEVADA
Carson City
IDAHO
Boise
MONTANA
Helena
WYOMING
Cheyenne
UTAH
Salt Lake City
COLORADO
Denver
ARIZONA
Phoenix
NEW MEXICO
Santa Fe
NORTH DAKOTA
Bismarck
SOUTH DAKOTA
Pierre
NEBRASKA
Lincoln
KANSAS
Topeka
OKLAHOMA
Oklahoma City
TEXAS
Austin
MINNESOTA
St Paul
IOWA
Des Moines
MISSOURI
Jefferson City
ARKANSAS
Little Rock
LOUISIANA
Baton Rouge
WISCONSIN
Madison
ILLINOIS
Springfield
MICHIGAN
Lansing
INDIANA
Indianapolis
OHIO
Columbus
KENTUCKY
Frankfort
TENNESSEE
Nashville
MISSISSIPPI
Jackson
ALABAMA
Montgomery
GEORGIA
Atlanta
FLORIDA
Tallahassee
SOUTH CAROLINA
Columbia
NORTH CAROLINA
Raleigh
WEST VIRGINIA
Charleston
VIRGINIA
Richmond
PENNSYLVANIA
Harrisburg
NEW YORK
Albany
MAINE
Augusta
VERMONT
Montpelier
NEW HAMPSHIRE
Concord
MASSACHUSETTS
Boston
RHODE ISLAND
Providence
CONNECTICUT
Hartford
NEW JERSEY
Trenton
DELAWARE
Dover
MARYLAND
Annapolis
DC
Washington
HAWAII
Honolulu
ALASKA
Juneau
PACIFIC OCEAN
ATLANTIC OCEAN
ARCTIC OCEAN
GULF OF MEXICO
MEXICO
CANADA
CUBA
THE BAHAMAS
BERING SEA
GULF OF ALASKA
RUSSIA
Lake Superior
Lake Michigan
Lake Huron
0 100 mi
0 100 km
0 100 200 300 mi
0 100 200 300 km
0 200 mi
0 200 km

Travelling Sales Person Tree

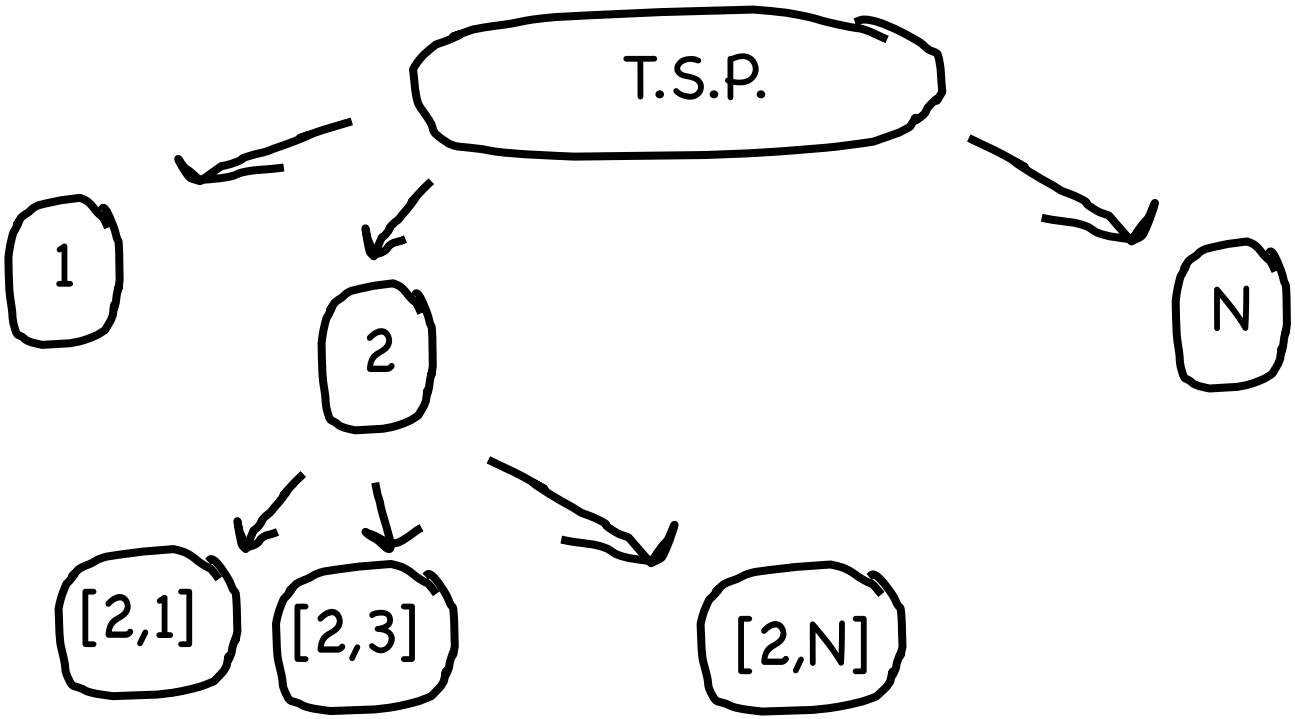


Travelling Sales Person Tree



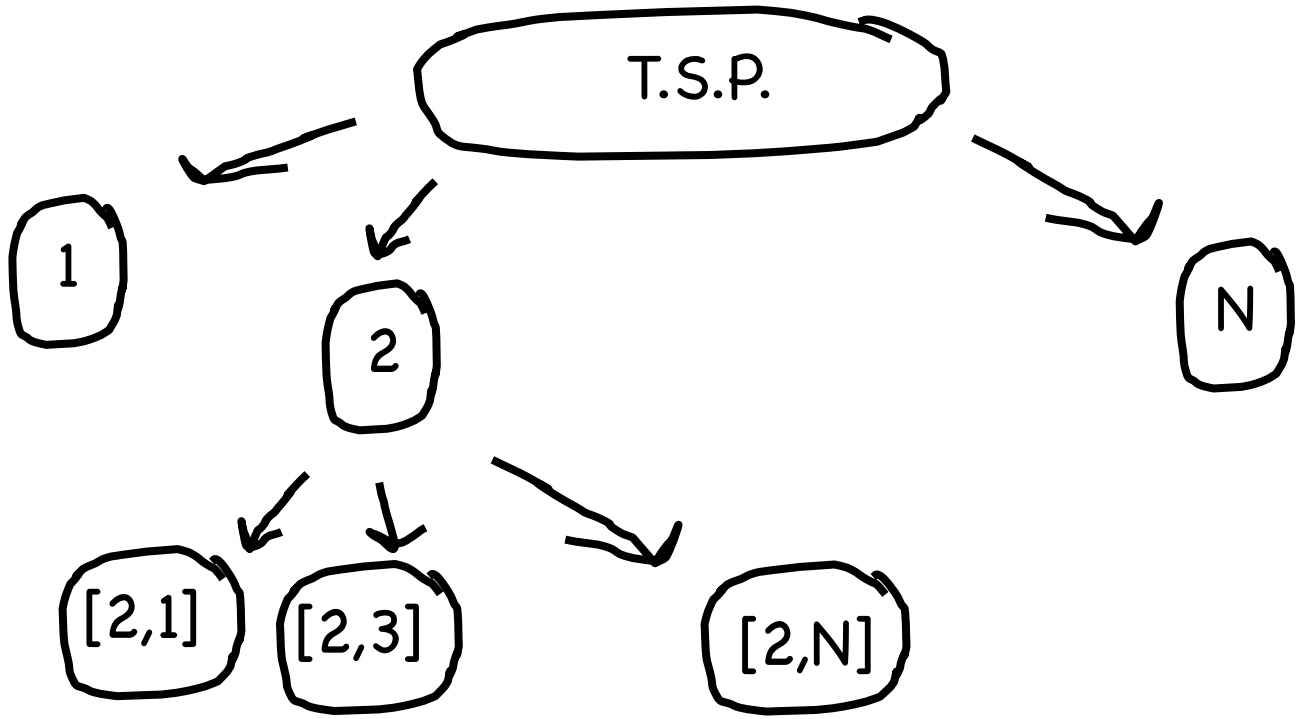
$$\# \, States = (N)$$

Travelling Sales Person Tree



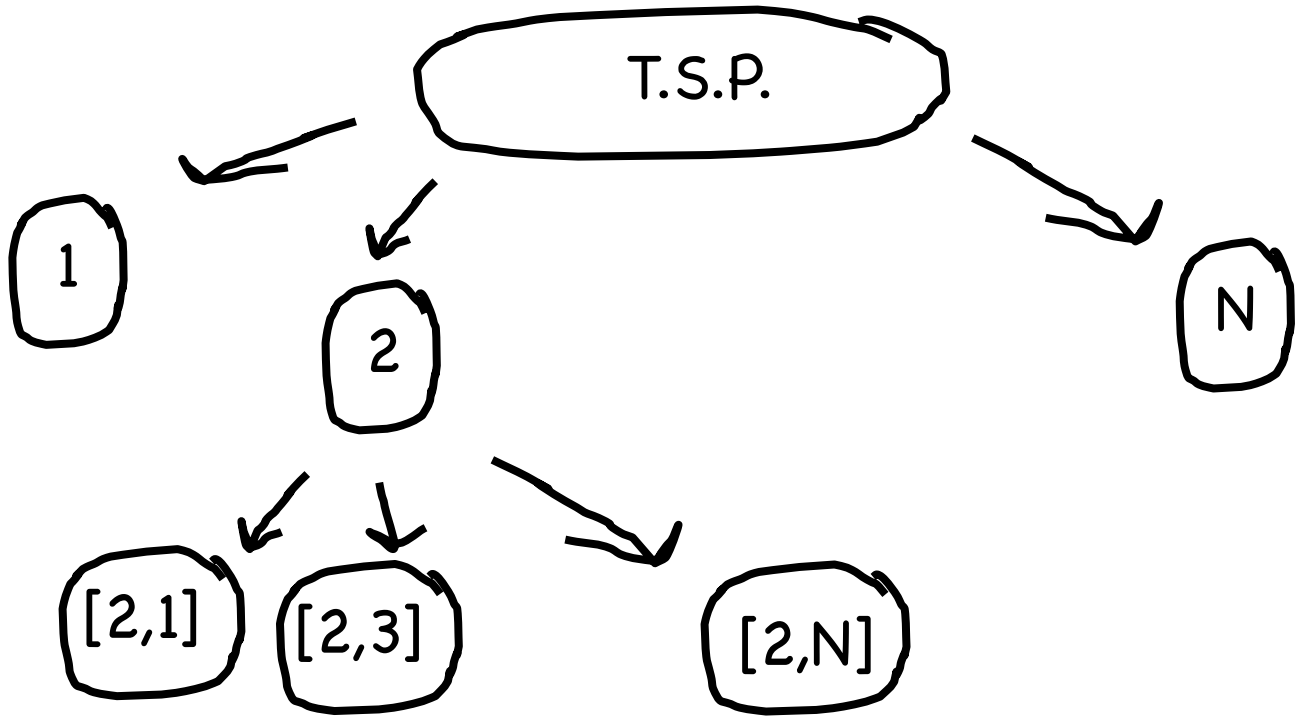
$$\# \, States = (N)\,(N-1)$$

Travelling Sales Person Tree



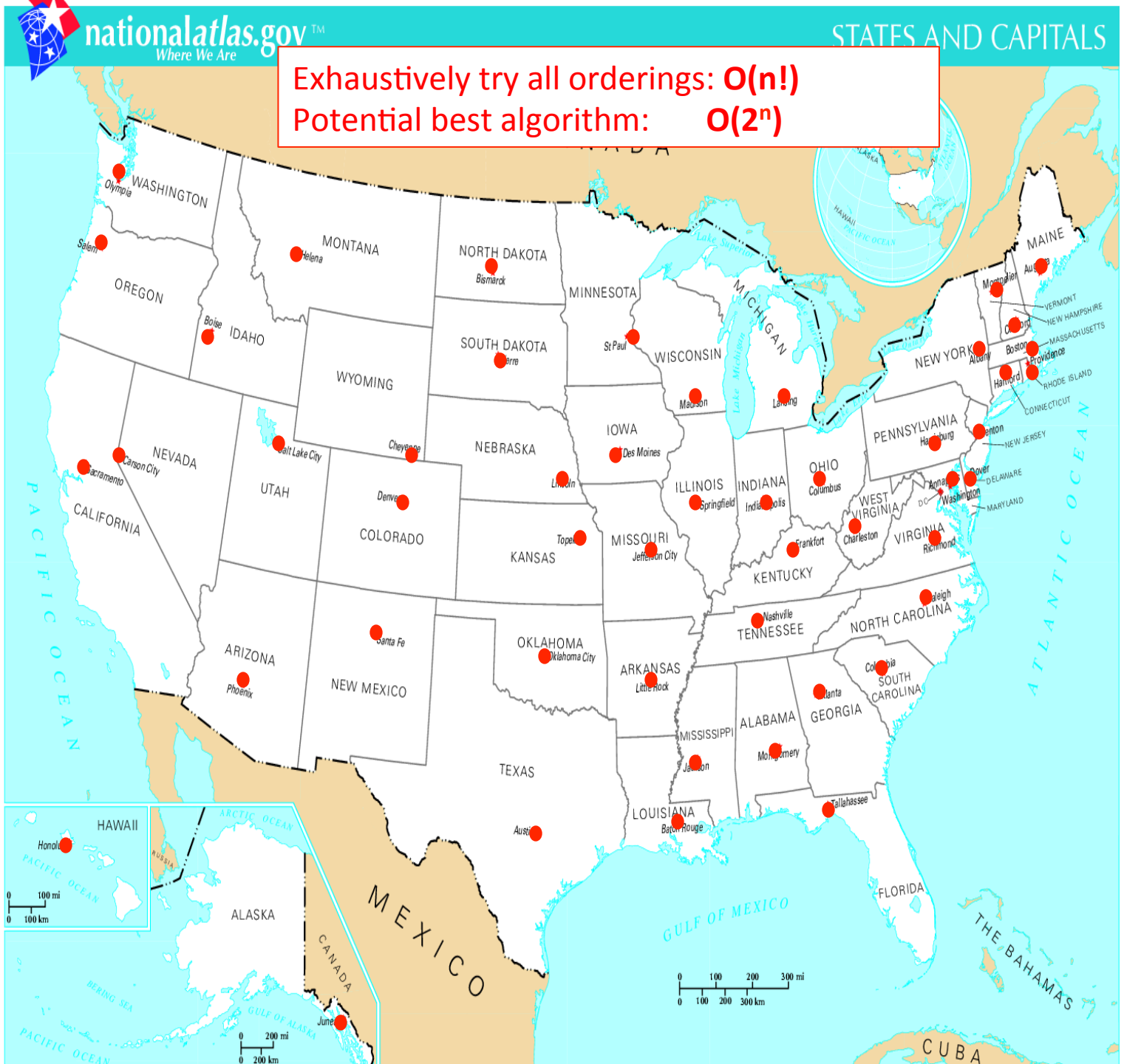
$$\# \, States = (N)\,(N-1) \times \ldots \times 1$$

Travelling Sales Person Tree



$$\# \, States = N!$$


nationalatlas.gov™
STATES AND CAPITALS
Traveling Salesperson Problem:
We have a bunch of cities to visit. In what order should we visit them to minimize total travel distance?
WASHINGTON
Olympia
OREGON
Salem
CALIFORNIA
Sacramento
NEVADA
Carson City
IDAHO
Boise
MONTANA
Helena
WYOMING
Cheyenne
UTAH
Salt Lake City
COLORADO
Denver
ARIZONA
Phoenix
NEW MEXICO
Santa Fe
NORTH DAKOTA
Bismarck
SOUTH DAKOTA
Pierre
NEBRASKA
Lincoln
KANSAS
Topeka
OKLAHOMA
Oklahoma City
TEXAS
Austin
MINNESOTA
St Paul
IOWA
Des Moines
MISSOURI
Jefferson City
ARKANSAS
Little Rock
LOUISIANA
Baton Rouge
WISCONSIN
Madison
ILLINOIS
Springfield
MICHIGAN
Lansing
INDIANA
Indianapolis
OHIO
Columbus
KENTUCKY
Frankfort
TENNESSEE
Nashville
MISSISSIPPI
Jackson
ALABAMA
Montgomery
GEORGIA
Atlanta
FLORIDA
Tallahassee
SOUTH CAROLINA
Columbia
NORTH CAROLINA
Raleigh
WEST VIRGINIA
Charleston
VIRGINIA
Richmond
PENNSYLVANIA
Harrisburg
NEW YORK
Albany
MAINE
Augusta
VERMONT
Montpelier
NEW HAMPSHIRE
Concord
MASSACHUSETTS
Boston
RHODE ISLAND
Providence
CONNECTICUT
Hartford
NEW JERSEY
Trenton
DELAWARE
Dover
MARYLAND
Annapolis
DC
Washington
HAWAII
Honolulu
ALASKA
Juneau
PACIFIC OCEAN
ATLANTIC OCEAN
ARCTIC OCEAN
GULF OF MEXICO
MEXICO
CANADA
CUBA
THE BAHAMAS
BERING SEA
GULF OF ALASKA
RUSSIA
Lake Superior
Lake Michigan
Lake Huron
0 100 mi
0 100 km
0 100 200 300 mi
0 100 200 300 km
0 200 mi
0 200 km

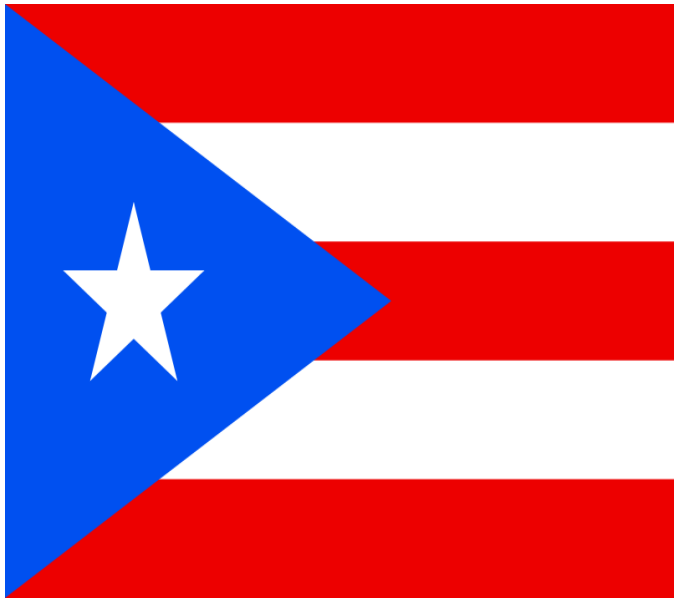

nationalatlas.gov
Where We Are
STATES AND CAPITALS
Exhaustively try all orderings: O(n!)
Potential best algorithm: O(2^n)
WASHINGTON
Olympia
OREGON
Salem
CALIFORNIA
Sacramento
NEVADA
Carson City
IDAHO
Boise
MONTANA
Helena
WYOMING
Cheyenne
UTAH
Salt Lake City
COLORADO
Denver
ARIZONA
Phoenix
NEW MEXICO
Santa Fe
NORTH DAKOTA
Bismarck
SOUTH DAKOTA
Pierre
NEBRASKA
Lincoln
KANSAS
Topeka
OKLAHOMA
Oklahoma City
TEXAS
Austin
MINNESOTA
St Paul
IOWA
Des Moines
MISSOURI
Jefferson City
ARKANSAS
Little Rock
LOUISIANA
Baton Rouge
WISCONSIN
Madison
ILLINOIS
Springfield
MICHIGAN
Lansing
INDIANA
Indianapolis
OHIO
Columbus
KENTUCKY
Frankfort
TENNESSEE
Nashville
MISSISSIPPI
Jackson
ALABAMA
Montgomery
GEORGIA
Atlanta
FLORIDA
Tallahassee
SOUTH CAROLINA
Columbia
NORTH CAROLINA
Raleigh
VIRGINIA
Richmond
WEST VIRGINIA
Charleston
PENNSYLVANIA
Harrisburg
NEW YORK
Albany
MAINE
Augusta
VERMONT
Montpelier
NEW HAMPSHIRE
Concord
MASSACHUSETTS
Boston
RHODE ISLAND
Providence
CONNECTICUT
Hartford
NEW JERSEY
Trenton
DELAWARE
Dover
MARYLAND
Annapolis
DC
Washington
HAWAII
Honolulu
ALASKA
Juneau
PACIFIC OCEAN
ATLANTIC OCEAN
ARCTIC OCEAN
GULF OF MEXICO
MEXICO
CANADA
CUBA
THE BAHAMAS
BERING SEA
GULF OF ALASKA
RUSSIA
Lake Superior
Lake Michigan
Lake Huron
0 100 mi
0 100 km
0 100 200 300 mi
0 100 200 300 km
0 200 mi
0 200 km

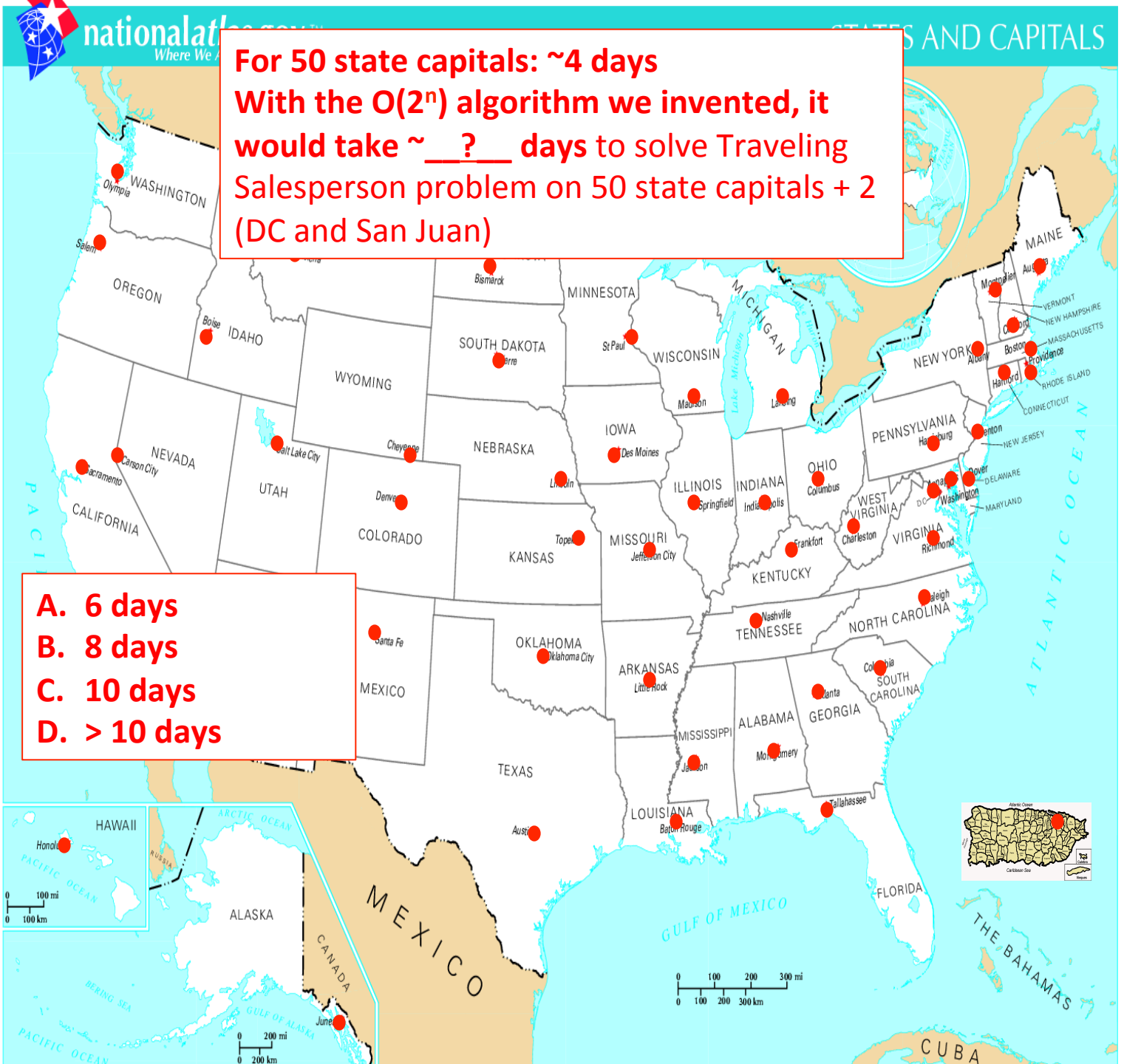

So let's say we come up with a way to solve Traveling Salesperson Problem in O(2^n).
It would take ~4 days to solve Traveling Salesperson Problem on 50 state capitals (with 3GHz computer)
nationalatlas.gov
STATES AND CAPITALS
WASHINGTON
Olympia
Salem
OREGON
Boise
IDAHO
WYOMING
Cheyenne
SOUTH DAKOTA
Pierre
St Paul
WISCONSIN
Madison
Lansing
MICHIGAN
NEW YORK
Albany
Boston
Providence
Hartford
MAINE
VERMONT
NEW HAMPSHIRE
MASSACHUSETTS
RHODE ISLAND
CONNECTICUT
NEVADA
Carson City
Sacramento
CALIFORNIA
Salt Lake City
UTAH
Denver
COLORADO
NEBRASKA
Lincoln
IOWA
Des Moines
ILLINOIS
Springfield
INDIANA
Indianapolis
OHIO
Columbus
PENNSYLVANIA
Harrisburg
Trenton
NEW JERSEY
Dover
DELAWARE
Annapolis
Washington
DC
MARYLAND
WEST VIRGINIA
Charleston
VIRGINIA
Richmond
KANSAS
Topeka
MISSOURI
Jefferson City
Frankfort
KENTUCKY
Nashville
TENNESSEE
Raleigh
NORTH CAROLINA
Columbia
SOUTH CAROLINA
Atlanta
GEORGIA
ALABAMA
Montgomery
MISSISSIPPI
Jackson
Tallahassee
FLORIDA
ARIZONA
Phoenix
Santa Fe
NEW MEXICO
OKLAHOMA
Oklahoma City
ARKANSAS
Little Rock
LOUISIANA
Baton Rouge
TEXAS
Austin
MEXICO
GULF OF MEXICO
ATLANTIC OCEAN
PACIFIC OCEAN
HAWAII
Honolulu
ALASKA
Juneau
CANADA
ARCTIC OCEAN
BERING SEA
GULF OF ALASKA
RUSSIA
THE BAHAMAS
CUBA
0 100 mi
0 100 km
0 200 mi
0 200 km
0 100 200 300 mi
0 100 200 300 km

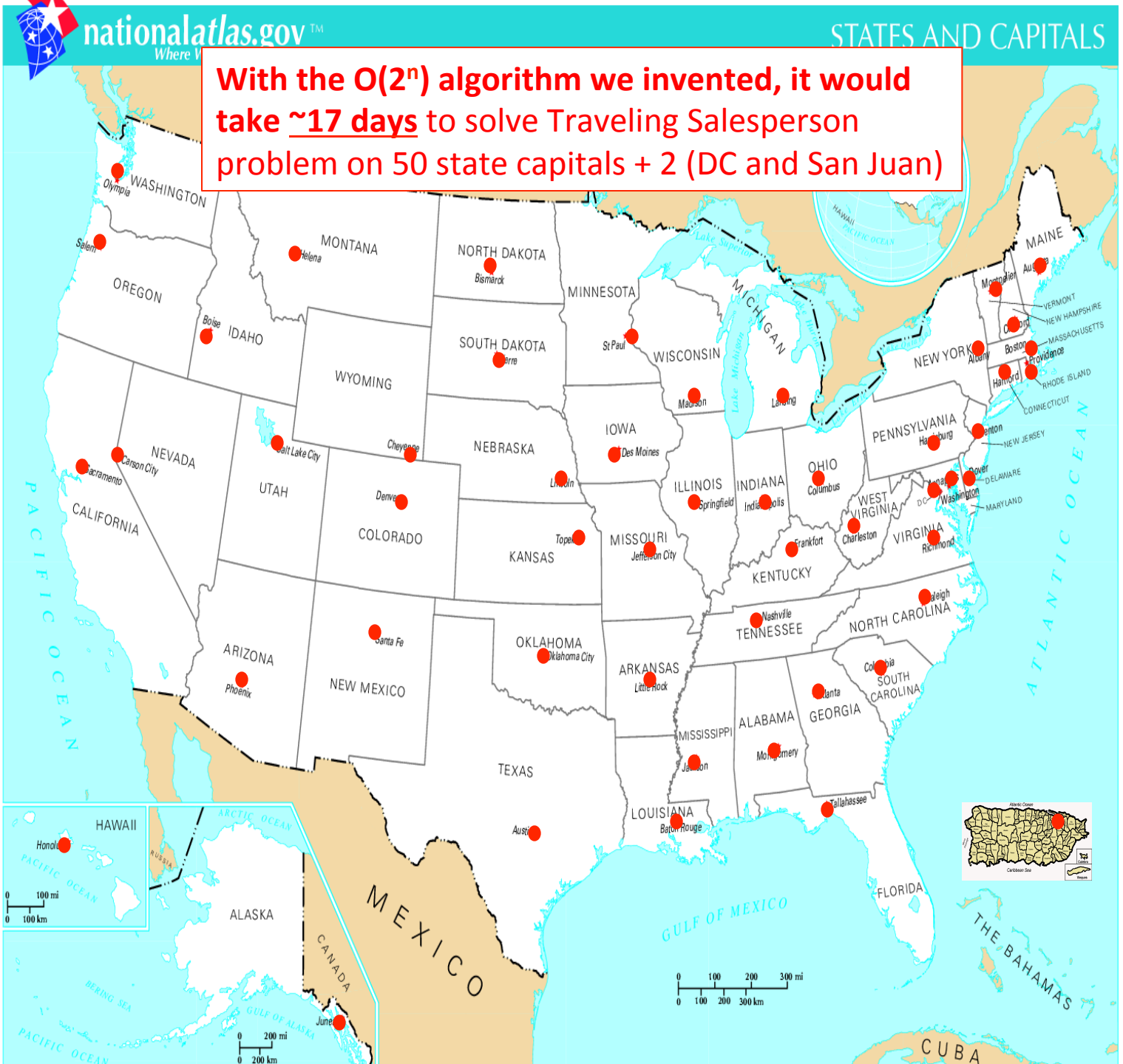
Two Tiny Updates

- Add San Juan, the capital city of Puerto Rico
- Also add Washington, DC

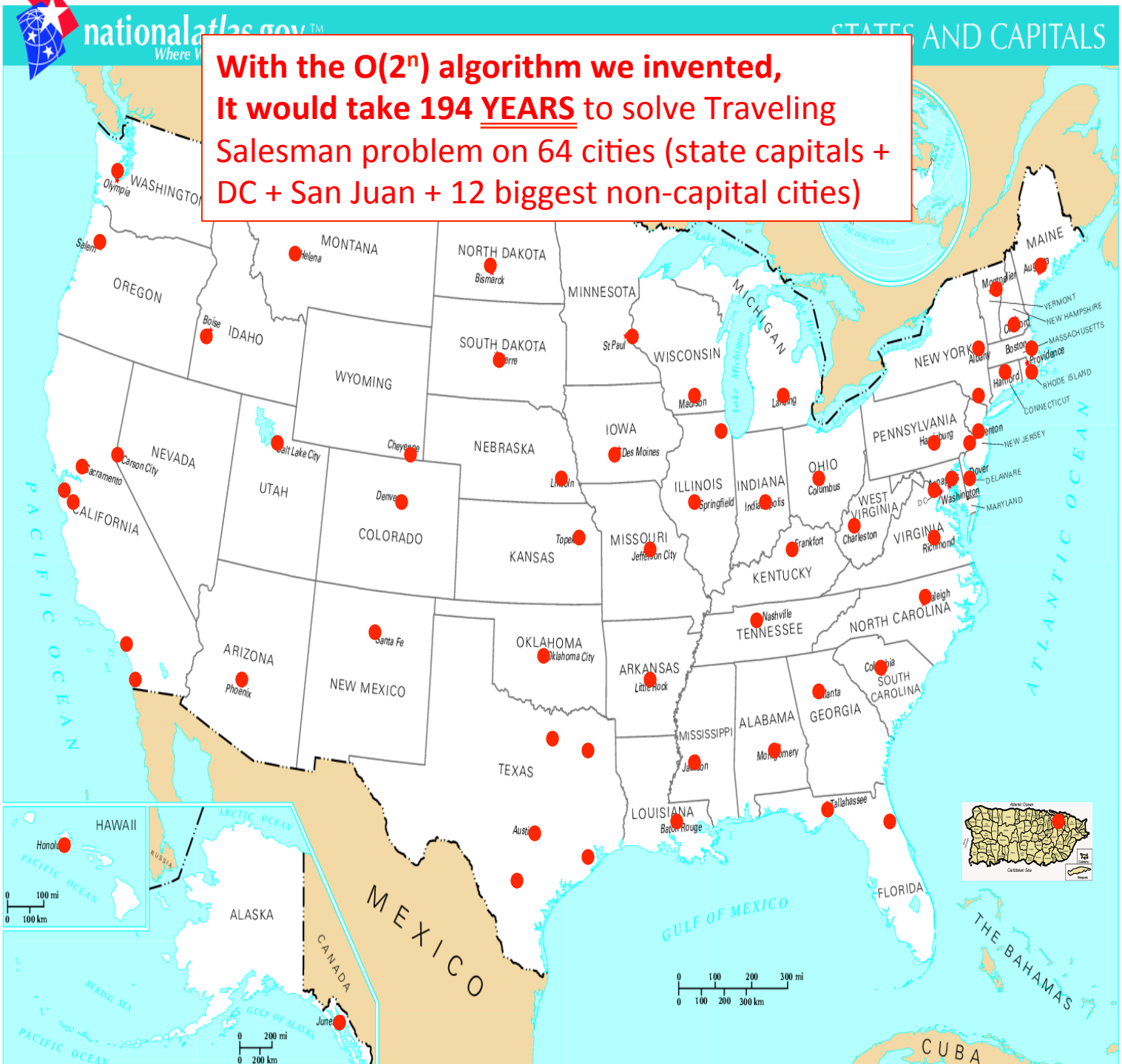
- **Now 52 capital cities instead of 50**




For 50 state capitals: ~4 days
With the O(2^n) algorithm we invented, it would take ~__?__ days to solve Traveling Salesperson problem on 50 state capitals + 2 (DC and San Juan)
A. 6 days
B. 8 days
C. 10 days
D. > 10 days
AND CAPITALS
WASHINGTON
Olympia
OREGON
Salem
IDAHO
Boise
NEVADA
Carson City
Sacramento
CALIFORNIA
UTAH
Salt Lake City
WYOMING
Cheyenne
COLORADO
Denver
NEBRASKA
Lincoln
KANSAS
Topeka
OKLAHOMA
Oklahoma City
TEXAS
Austin
Santa Fe
MEXICO
SOUTH DAKOTA
Pierre
Bismarck
MINNESOTA
St Paul
IOWA
Des Moines
MISSOURI
Jefferson City
ARKANSAS
Little Rock
LOUISIANA
Baton Rouge
WISCONSIN
Madison
ILLINOIS
Springfield
MICHIGAN
Lansing
INDIANA
Indianapolis
OHIO
Columbus
KENTUCKY
Frankfort
TENNESSEE
Nashville
MISSISSIPPI
Jackson
ALABAMA
Montgomery
GEORGIA
Atlanta
FLORIDA
Tallahassee
SOUTH CAROLINA
Columbia
NORTH CAROLINA
Raleigh
WEST VIRGINIA
Charleston
VIRGINIA
Richmond
PENNSYLVANIA
Harrisburg
NEW YORK
Albany
MAINE
Augusta
VERMONT
Montpelier
NEW HAMPSHIRE
Concord
MASSACHUSETTS
Boston
RHODE ISLAND
Providence
CONNECTICUT
Hartford
NEW JERSEY
Trenton
DELAWARE
Dover
MARYLAND
DC
Washington
HAWAII
Honolulu
ALASKA
Juneau
CANADA
RUSSIA
ARCTIC OCEAN
BERING SEA
GULF OF ALASKA
PACIFIC OCEAN
GULF OF MEXICO
ATLANTIC OCEAN
THE BAHAMAS
CUBA
Caribbean Sea
0 100 mi
0 100 km
0 200 mi
0 200 km
0 100 200 300 mi
0 100 200 300 km


nationalatlas.gov
STATES AND CAPITALS
With the O(2^n) algorithm we invented, it would take ~17 days to solve Traveling Salesperson problem on 50 state capitals + 2 (DC and San Juan)
WASHINGTON
Olympia
OREGON
Salem
CALIFORNIA
Sacramento
NEVADA
Carson City
IDAHO
Boise
MONTANA
Helena
WYOMING
Cheyenne
UTAH
Salt Lake City
COLORADO
Denver
ARIZONA
Phoenix
NEW MEXICO
Santa Fe
NORTH DAKOTA
Bismarck
SOUTH DAKOTA
Pierre
NEBRASKA
Lincoln
KANSAS
Topeka
OKLAHOMA
Oklahoma City
TEXAS
Austin
MINNESOTA
St Paul
IOWA
Des Moines
MISSOURI
Jefferson City
ARKANSAS
Little Rock
LOUISIANA
Baton Rouge
WISCONSIN
Madison
ILLINOIS
Springfield
MICHIGAN
Lansing
INDIANA
Indianapolis
OHIO
Columbus
KENTUCKY
Frankfort
TENNESSEE
Nashville
MISSISSIPPI
Jackson
ALABAMA
Montgomery
GEORGIA
Atlanta
FLORIDA
Tallahassee
SOUTH CAROLINA
Columbia
NORTH CAROLINA
Raleigh
VIRGINIA
Richmond
WEST VIRGINIA
Charleston
PENNSYLVANIA
Harrisburg
NEW YORK
Albany
MAINE
Augusta
VERMONT
Montpelier
NEW HAMPSHIRE
Concord
MASSACHUSETTS
Boston
RHODE ISLAND
Providence
CONNECTICUT
Hartford
NEW JERSEY
Trenton
DELAWARE
Dover
MARYLAND
Annapolis
DC
Washington
HAWAII
Honolulu
ALASKA
Juneau
PACIFIC OCEAN
ATLANTIC OCEAN
ARCTIC OCEAN
GULF OF MEXICO
MEXICO
CANADA
RUSSIA
BERING SEA
GULF OF ALASKA
THE BAHAMAS
CUBA
Lake Superior
Lake Michigan
Lake Huron
Atlantic Ocean
Caribbean Sea
0 100 mi
0 100 km
0 100 200 300 mi
0 100 200 300 km
0 200 mi
0 200 km


nationalatlas.gov
STATES AND CAPITALS
Sacramento is not exactly the most interesting or important city in California (sorry, Sacramento). **What if we add the 12 biggest non-capital cities in the United States to our map?**
WASHINGTON
Olympia
OREGON
Salem
IDAHO
Boise
NEVADA
Carson City
Sacramento
CALIFORNIA
UTAH
Salt Lake City
WYOMING
Cheyenne
COLORADO
Denver
NEBRASKA
Lincoln
KANSAS
Topeka
OKLAHOMA
Oklahoma City
TEXAS
Austin
NEW MEXICO
Santa Fe
ARIZONA
Phoenix
SOUTH DAKOTA
Pierre
Bismarck
MINNESOTA
St Paul
IOWA
Des Moines
MISSOURI
Jefferson City
ARKANSAS
Little Rock
LOUISIANA
Baton Rouge
WISCONSIN
Madison
ILLINOIS
Springfield
MICHIGAN
Lansing
INDIANA
Indianapolis
OHIO
Columbus
KENTUCKY
Frankfort
TENNESSEE
Nashville
MISSISSIPPI
Jackson
ALABAMA
Montgomery
GEORGIA
Atlanta
FLORIDA
Tallahassee
SOUTH CAROLINA
Columbia
NORTH CAROLINA
Raleigh
WEST VIRGINIA
Charleston
VIRGINIA
Richmond
PENNSYLVANIA
Harrisburg
NEW YORK
Albany
MAINE
Augusta
VERMONT
Montpelier
NEW HAMPSHIRE
Concord
MASSACHUSETTS
Boston
RHODE ISLAND
Providence
CONNECTICUT
Hartford
NEW JERSEY
Trenton
DELAWARE
Dover
MARYLAND
Annapolis
DC
Washington
PACIFIC OCEAN
ATLANTIC OCEAN
GULF OF MEXICO
MEXICO
CANADA
CUBA
THE BAHAMAS
HAWAII
Honolulu
ALASKA
Juneau
ARCTIC OCEAN
RUSSIA
BERING SEA
GULF OF ALASKA
Lake Michigan
Atlantic Ocean
Caribbean Sea
Virgin Islands
Vieques
0 100 mi
0 100 km
0 100 200 300 mi
0 100 200 300 km
0 200 mi
0 200 km


With the O(2^n) algorithm we invented,
It would take 194 YEARS to solve Traveling Salesman problem on 64 cities (state capitals + DC + San Juan + 12 biggest non-capital cities)
nationalatlas.gov
STATES AND CAPITALS
WASHINGTON
Olympia
OREGON
Salem
CALIFORNIA
Sacramento
NEVADA
Carson City
IDAHO
Boise
MONTANA
Helena
WYOMING
Cheyenne
UTAH
Salt Lake City
COLORADO
Denver
ARIZONA
Phoenix
NEW MEXICO
Santa Fe
NORTH DAKOTA
Bismarck
SOUTH DAKOTA
Pierre
NEBRASKA
Lincoln
KANSAS
Topeka
OKLAHOMA
Oklahoma City
TEXAS
Austin
MINNESOTA
St Paul
IOWA
Des Moines
MISSOURI
Jefferson City
ARKANSAS
Little Rock
LOUISIANA
Baton Rouge
WISCONSIN
Madison
ILLINOIS
Springfield
MICHIGAN
Lansing
INDIANA
Indianapolis
OHIO
Columbus
KENTUCKY
Frankfort
TENNESSEE
Nashville
MISSISSIPPI
Jackson
ALABAMA
Montgomery
GEORGIA
Atlanta
FLORIDA
Tallahassee
SOUTH CAROLINA
Columbia
NORTH CAROLINA
Raleigh
VIRGINIA
Richmond
WEST VIRGINIA
Charleston
PENNSYLVANIA
Harrisburg
NEW YORK
Albany
MAINE
Augusta
VERMONT
Montpelier
NEW HAMPSHIRE
Concord
MASSACHUSETTS
Boston
RHODE ISLAND
Providence
CONNECTICUT
Hartford
NEW JERSEY
Trenton
DELAWARE
Dover
MARYLAND
Annapolis
DC
Washington
HAWAII
Honolulu
ALASKA
Juneau
PACIFIC OCEAN
ATLANTIC OCEAN
ARCTIC OCEAN
GULF OF MEXICO
MEXICO
CANADA
CUBA
THE BAHAMAS
RUSSIA
BERING SEA
GULF OF ALASKA
Lake Superior
Lake Michigan
Caribbean Sea
0 100 mi
0 100 km
0 200 mi
0 200 km
0 100 200 300 mi
0 100 200 300 km

$\log_2 n$	n	$n \log_2 n$	n^2	2^n
2	**4**	8	16	16
3	**8**	24	64	256
4	**16**	64	256	65,536
5	**32**	160	1,024	4,294,967,296
6	**64**	384	4,096	1.84×10^{19}
7	**128**			
8	**256**			
9	**512**			
10	**1,024**			
30	**1,300,000,000**			

# of Facebook accounts

$\log_2 n$	n	$n \log_2 n$	n^2	2^n
2	**4**	8	16	16
3	**8**	24	64	256
4	**16**	64	256	65,536
5	**32**	160	1,024	4,294,967,296
6	**64**	384	4,096	1.84×10^{19}
7	**128**			194 **YEARS**
8	**256**			
9	**512**			
10	**1,024**			
30	**1,300,000,000**			

# of Facebook accounts

$\log_2 n$	n	$n \log_2 n$	n^2	2^n
2	**4**	8	16	16
3	**8**	24	64	256
4	**16**	64	256	65,536
5	**32**	160	1,024	4,294,967,296
6	**64**	384	4,096	1.84×10^{19}
7	**128**	896	16,384	3.40×10^{38}
8	**256**			
9	**512**			
10	**1,024**			
30	**1,300,000,000**			

3,590,000,000,000,000,000,000
YEARS

# of Facebook accounts

$\log_2 n$	n	$n \log_2 n$	n^2	2^n
2	**4**	8	16	16
3	**8**	24	64	256
4	**16**	64	256	65,536
5	**32**	160	1,024	4,294,967,296
6	**64**	384	4,096	1.84×10^{19}
7	**128**	896	16,384	3.40×10^{38}
8	**256**	2,048	65,536	1.16×10^{77}
9	**512**			
10	**1,024**			
30	**1,600,000,000**			

For comparison: there are about 10E+80 atoms in the universe. No big deal.

# of Facebook accounts

$\log_2 n$	n	$n \log_2 n$	n^2	2^n
2	**4**	8	16	16
3	**8**	24	64	256
4	**16**	64	256	65,536
5	**32**	160	1,024	4,294,967,296
6	**64**	384	4,096	1.84×10^{19}
7	**128**	896	16,384	3.40×10^{38}
8	**256**	2,048	65,536	1.16×10^{77}
9	**512**	4,608	262,144	1.34×10^{154}
10	**1,024**			
30	**1,300,000,000**			

1.42E+137 **YEARS** (another way of thinking about the size: including commas, this number of years cannot be written in a single tweet)

# of Facebook accounts

$\log_2 n$	n	$n \log_2 n$	n^2	2^n
2	**4**	8	16	16
3	**8**	24	64	256
4	**16**	64	256	65,536
5	**32**	160	1,024	4,294,967,296
6	**64**	384	4,096	1.84×10^{19}
7	**128**	896	16,384	3.40×10^{38}
8	**256**	2,048	65,536	1.16×10^{77}
9	**512**	4,608	262,144	1.34×10^{154}
10	**1,024**	10,240 (.000003s)	1,048,576 (.0003s)	1.80×10^{308}
30	**1,300,000,000**	39000000000 (13s)	1690000000000000000 (18 years)	LOL

# of Facebook accounts

$\log_2 n$	n	$n \log_2 n$	n^2	2^n
2	**4**	8	16	16
3	**8**	24	64	256
4	**16**	64	256	65,536
5	**32**	160	1,024	4,294,967,296
6	**64**	384	4,096	1.84×10^{19}
7	**128**	896	16,384	3.40×10^{38}
8	**256**	2,048	65,536	1.16×10^{77}
9	**512**			$\times 10^{154}$
10	**1,024**	(.000003s)		$\times 10^{308}$
30	**1,300,000,000**	39000000000 (13s)	1690000000000000000 (18 years)	LOL

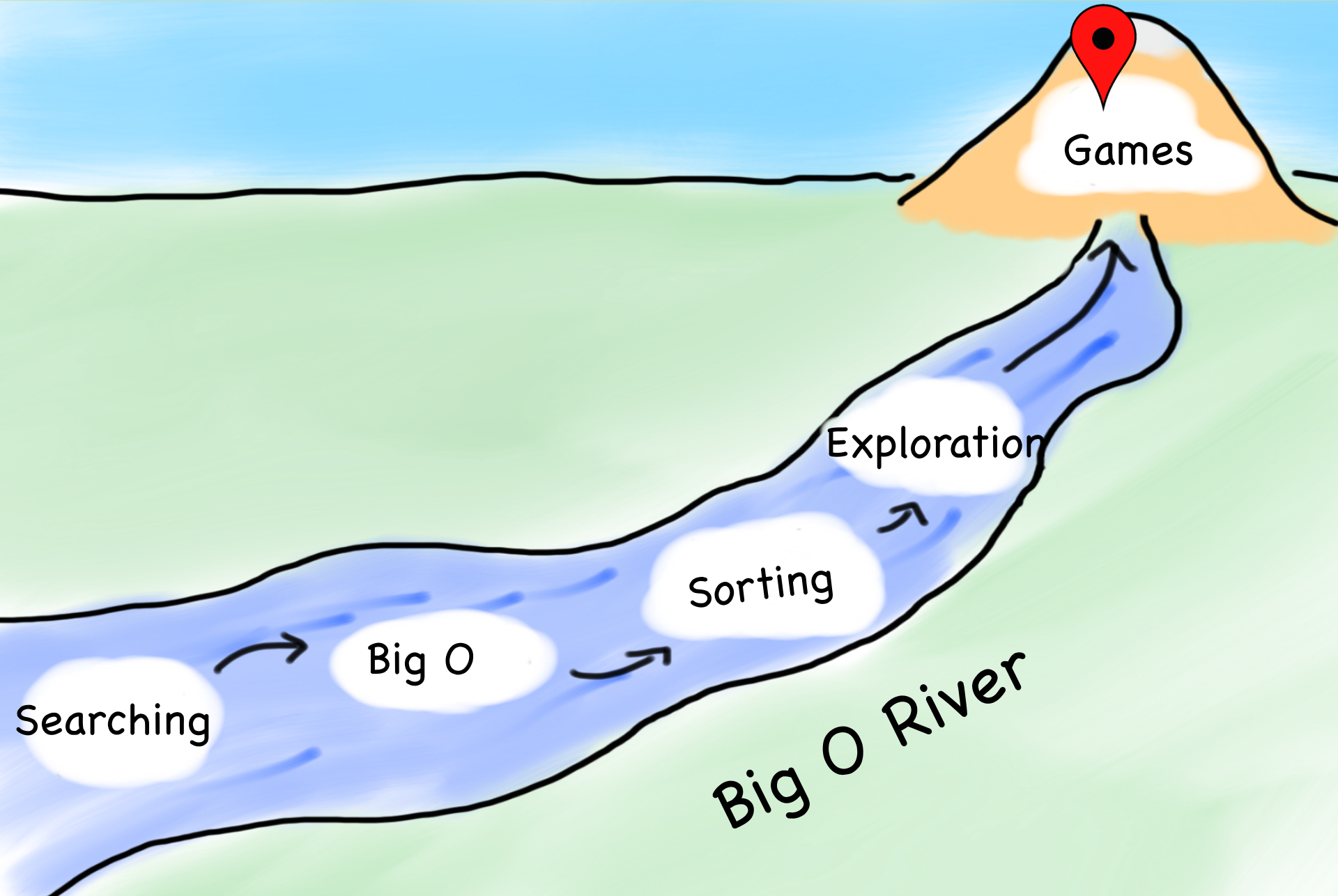
2^n is way into crazy LOL territory, but **look at $n\log_2 n$—only 13 seconds!!**

# of Facebook accounts


THIS IS ME NOT CARING
ABOUT PERFORMANCE TUNING UNLESS IT CHANGES BIG-O
memegenerator.net

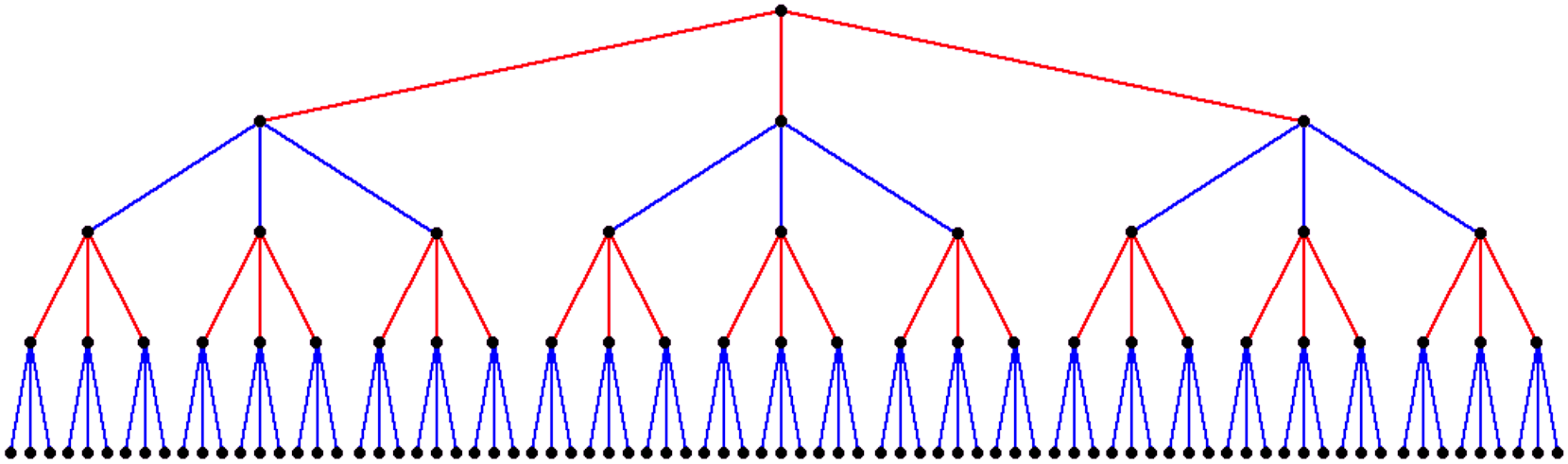
Today's Goals

Today's Goals



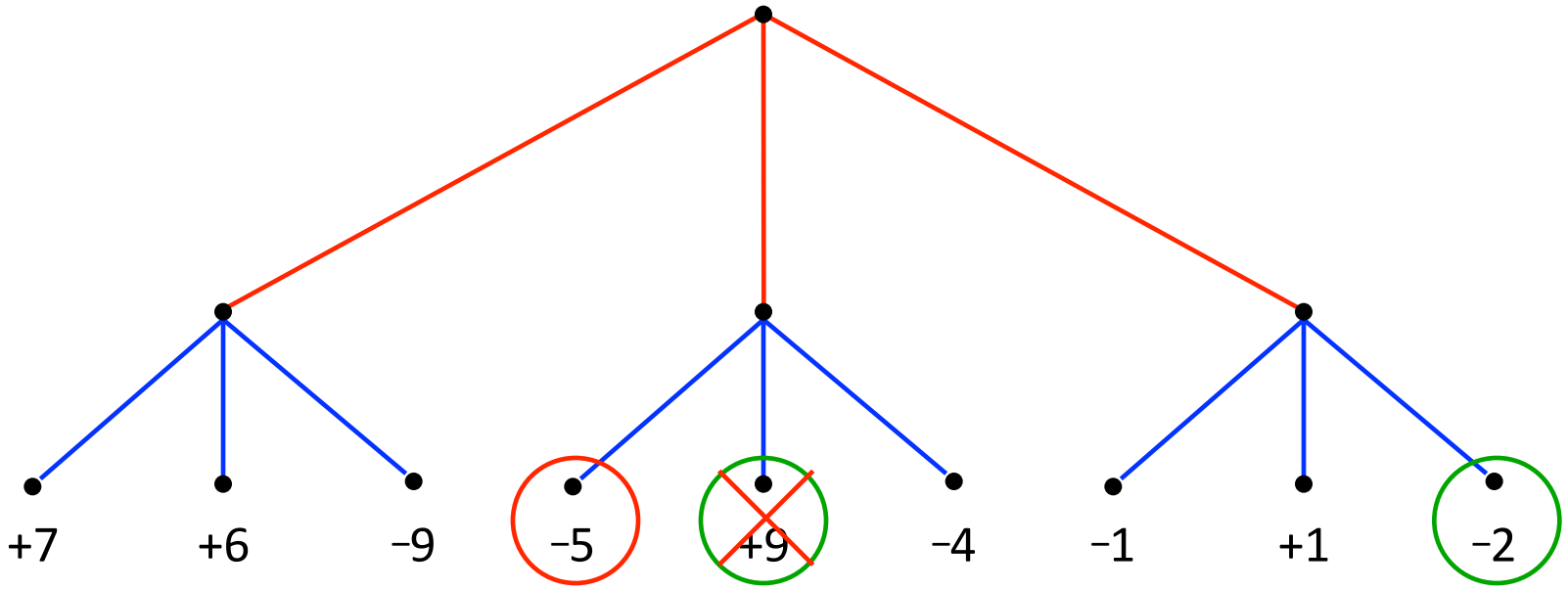
Game Trees

- As Shannon observed in 1950, most two-player games have the same basic form:
  - The first player (red) must choose between a set of moves
  - For each move, the second player (blue) has several responses.
  - For each of these responses, red has further choices.
  - For each of these new responses, blue makes another decision.
  - And so on . . .



Minimax Illustration

- Suppose that the ratings two turns from now are as shown.
- From your perspective, the +9 initially looks attractive.
- Unfortunately, your can't get there, since the –5 is better for your opponent.
- The best you can do is choose the move that leads to the –2.



The Game of Nim

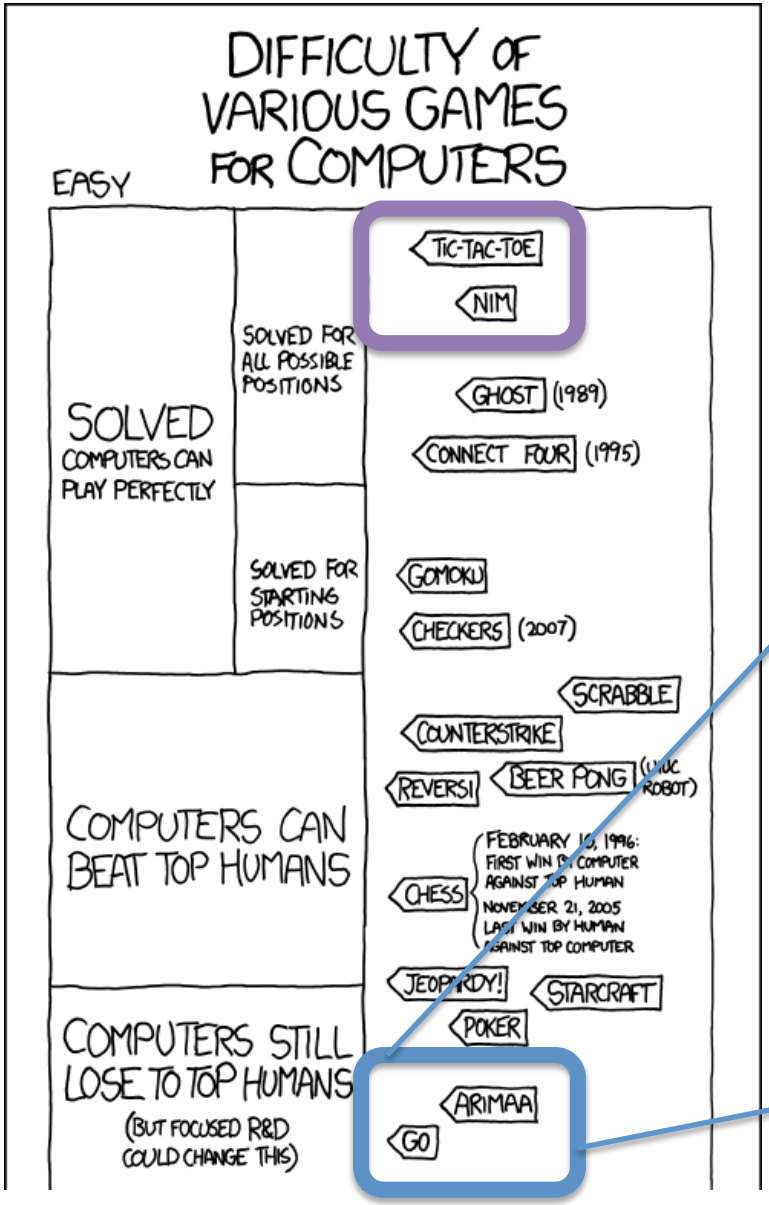
- In Nim, the game begins with a pile of coins between two players. The starting number of coins can vary and should therefore be easy to change in the program.
- In alternating turns, each player takes one, two, or three coins from the pile in the center.
- The player who takes the last coin loses.

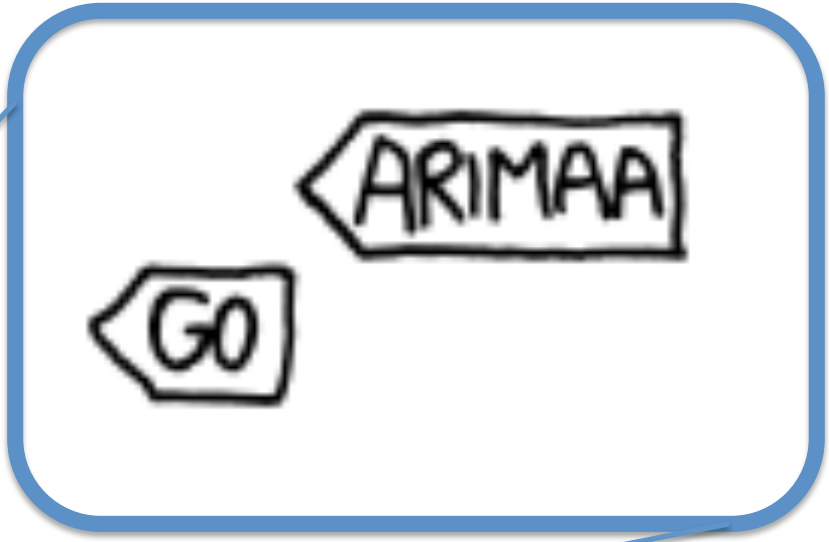
The Game of Nim

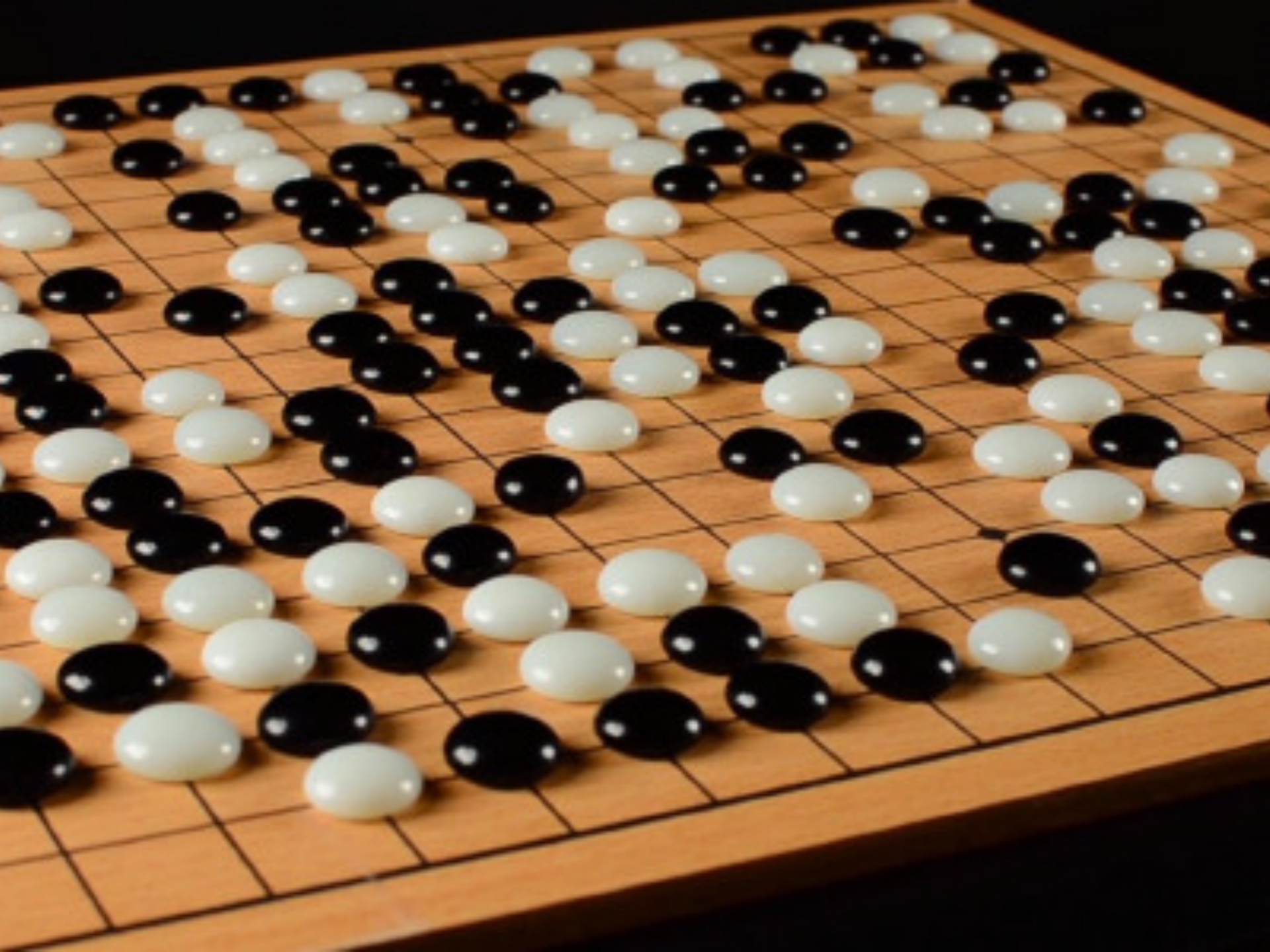
Nim

```
There are 11 coins left.
I'll take 2.
There are 9 coins left.
Your move: 1
There are 8 coins left.
I'll take 3.
There are 5 coins left.
Your move: 2
There are 3 coins left.
I'll take 2.
There is only one coin left.
You lose.
```

The Hardest Game Left







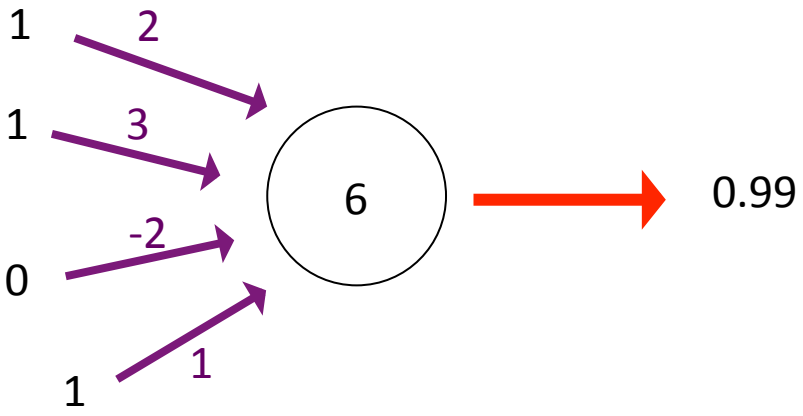
Go Complexity

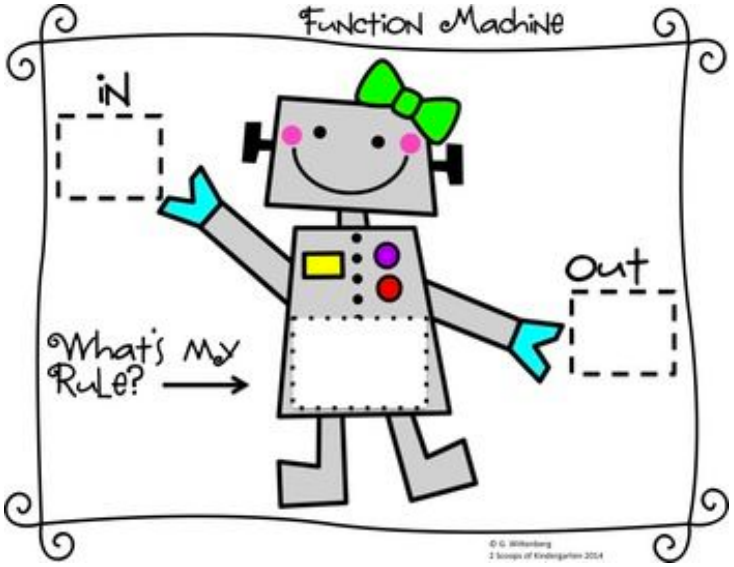
Complexity $\mathcal{O}(n^2!)$

When n=19 10^{768}


won
won

Remember This?



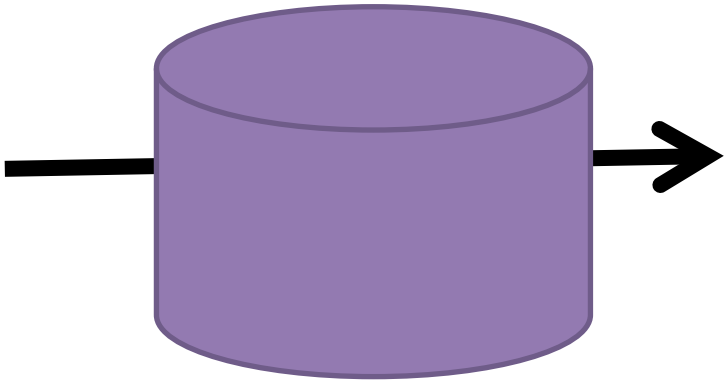



```
InputSum  = input1 * weight1 +
            input2 * weight2 +
            input3 * weight3 +
            input4 * weight4;
Output    = sigmoid(InputSum);
```

```
double neuron(double x1, double x2);
```

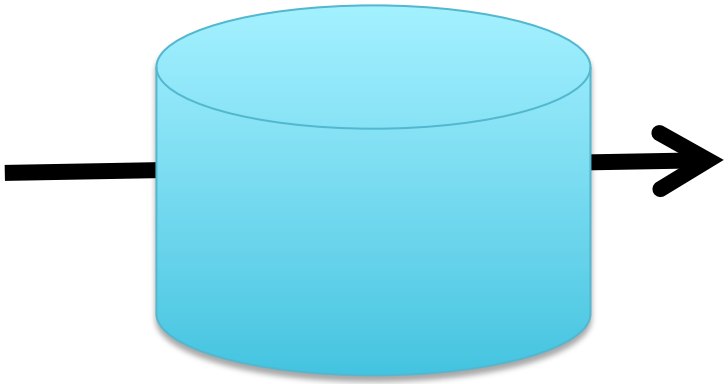
Two Neural Networks

Policy Network



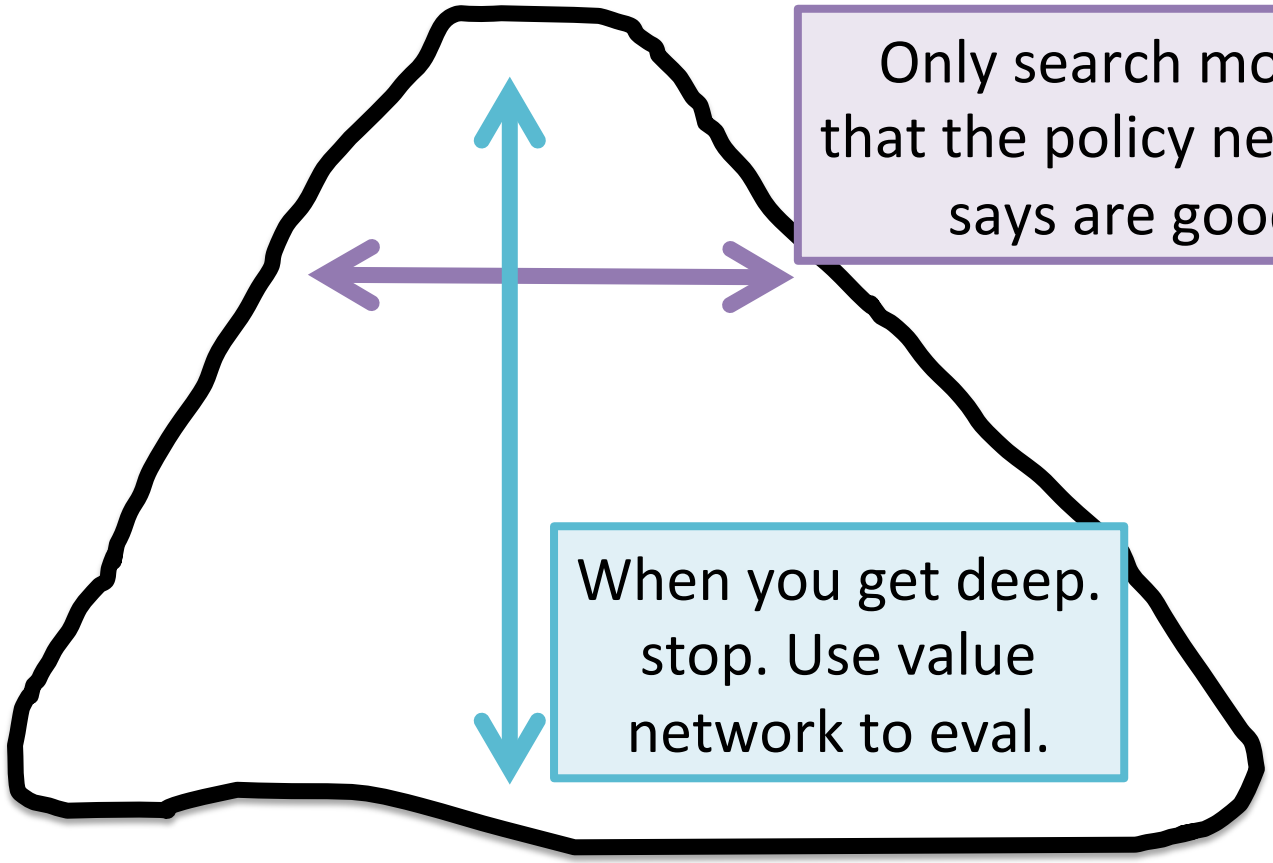
{Small set of moves}

Value Network



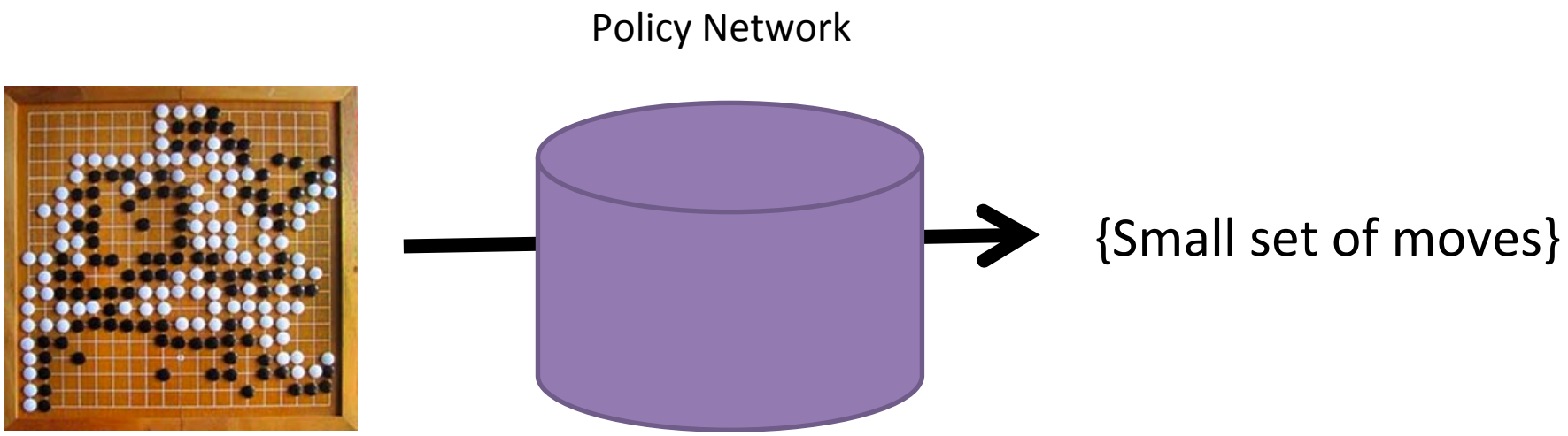
Score of board

Two Neural Networks



AlphaGo Big O

$$\mathcal{O}(n^2)^*$$



* But the constant factor is large

Today's Goals

1. Understand Big O
2. Get a feel for exponential growth