Collections, Part Two

Announcements

- Sections announced over email and room locations are now posted.
- Not in a section? Sign up for an open section at

http://cs198.stanford.edu/section

- Mac Users: We now have two versions of the starter files:
  - One version is purely for 10.7
  - The other version is for 10.6 and 10.8.
- When submitting, make sure to submit your .cpp source files, not the Xcode or Visual Studio project files!

Announcements

- Casual dinner for women studying CS **tonight** at 5:00PM at the Gates Patio.
- Everyone is welcome!
- RSVP through link sent out last Friday, or by visiting

http://bit.ly/casualcsdinner

she++

Vector

Vector

- The **`Vector`** is a collection class representing a list of things.
  - Similar to Java's `ArrayList` type.
- Probably the single most commonly used collection type in all programming.

Example: Cell Tower Purchasing

Buying Cell Towers

137 42 95 272 52

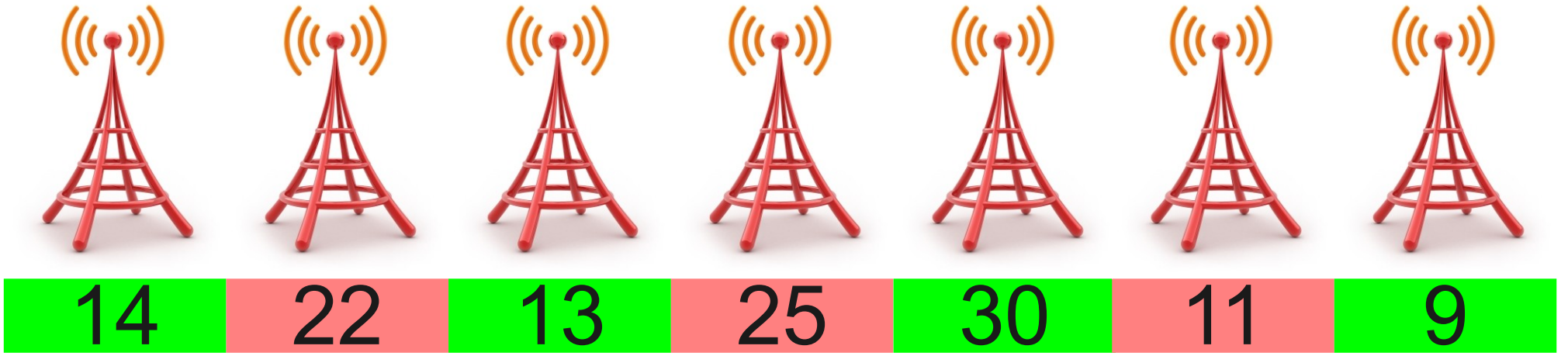
Buying Cell Towers

137	42	95	272	52

Buying Cell Towers

14 22 13 25 30 11 9

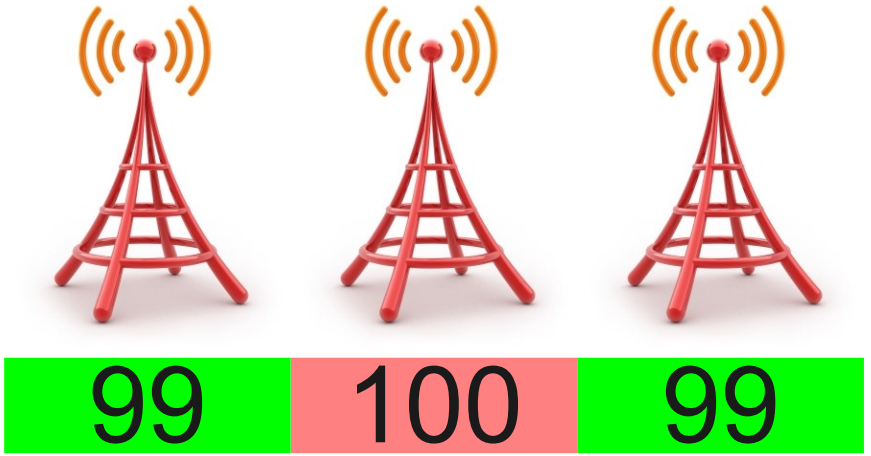
Buying Cell Towers



Buying Cell Towers

99 100 99

Buying Cell Towers



Given the populations of each city, what is the largest number of people you can provide service to?

Pass-by-Reference and Objects

- Recall: In C++, *all* parameters are passed by value unless specified otherwise.
- When using container types (**`Stack`**, **`Vector`**, etc.) it is often useful to use pass-by-reference for efficiency reasons.
  - Takes a *long* time to make a copy of a large collection!







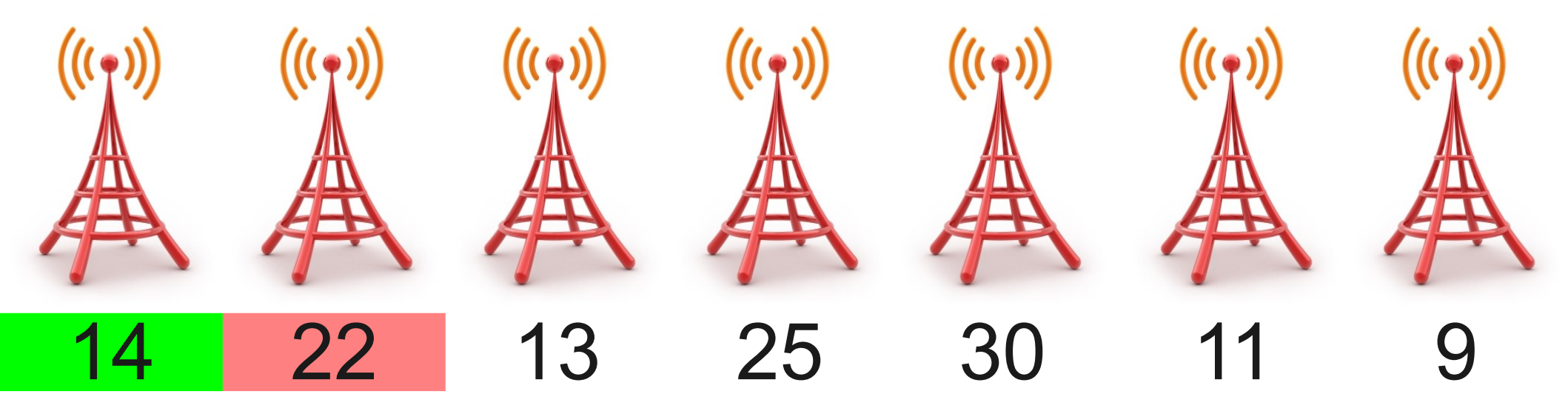
14 22 13 25 30 11 9

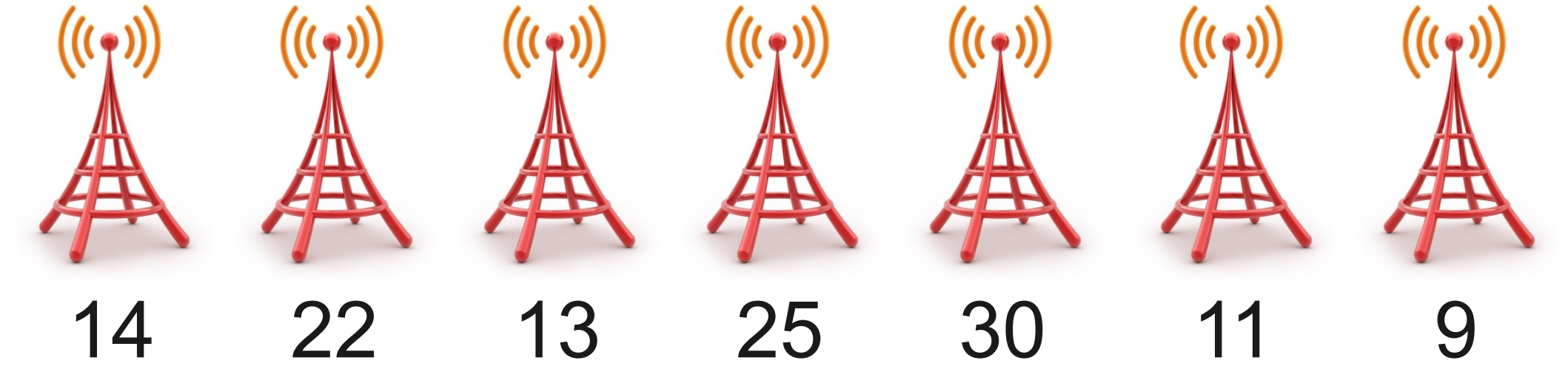
14	22	13	25	30	11	9

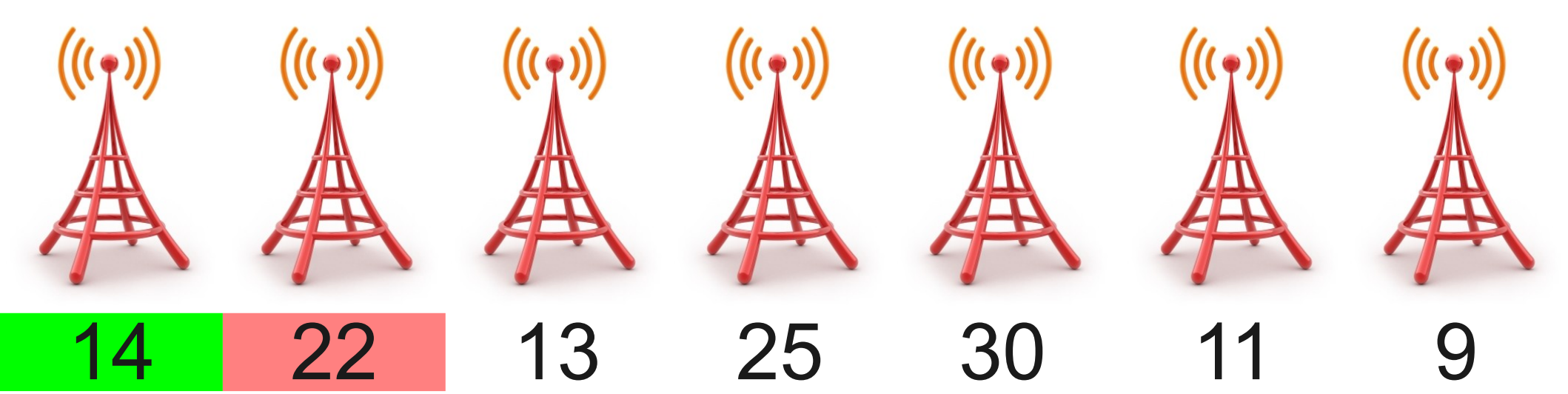
14	22	13	25	30	11	9

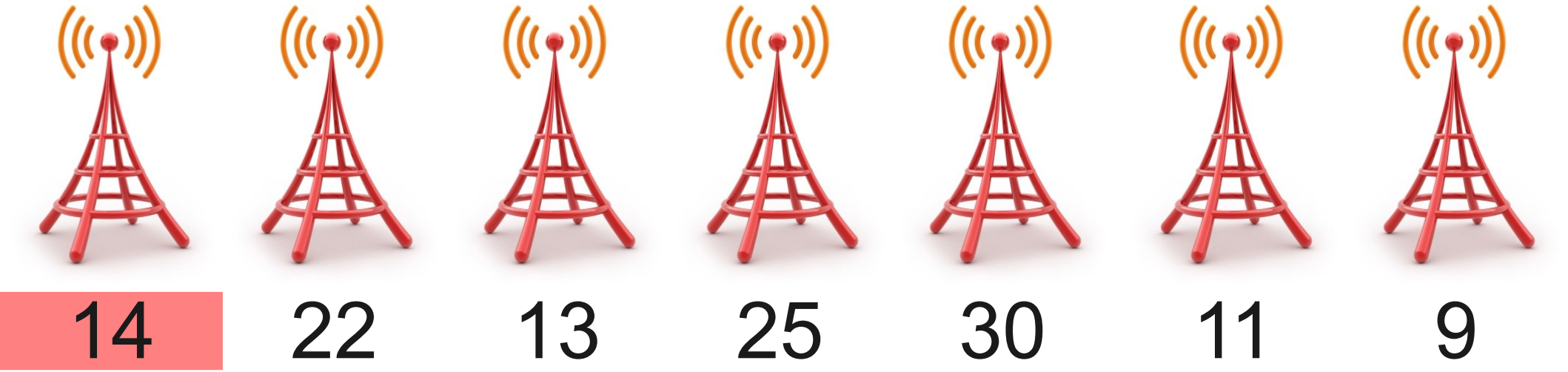
14 22 13 25 30 11 9

Maximize what's left in here.

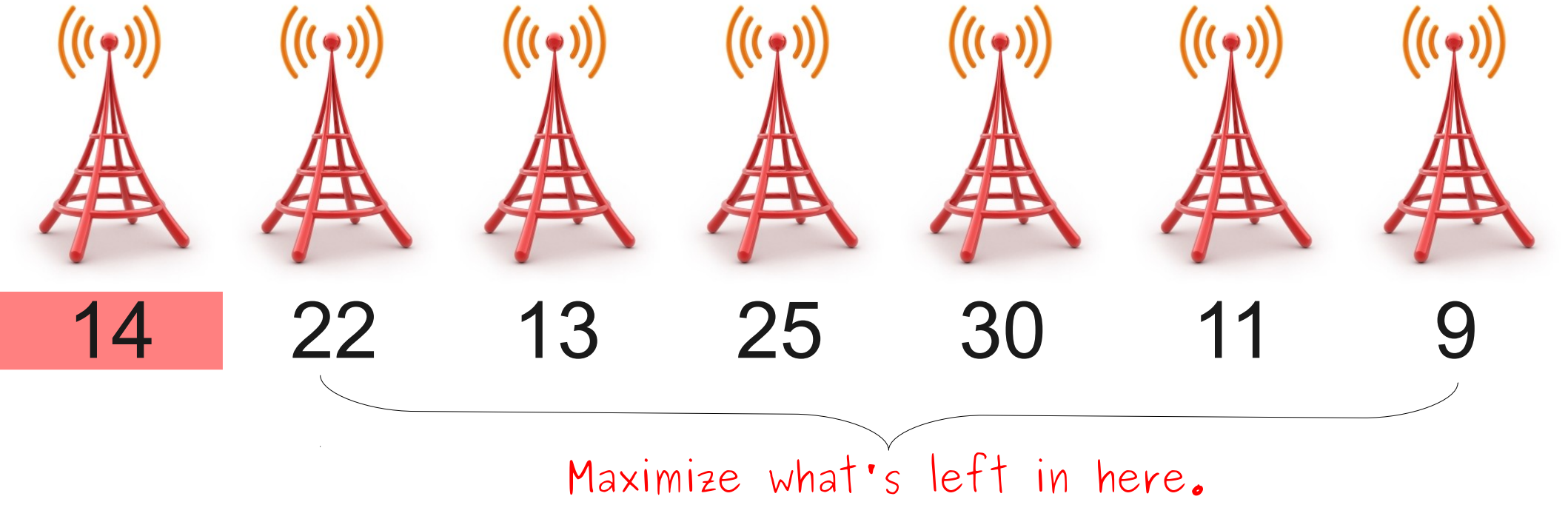

14
22
13
25
30
11
9
Maximize what's left in here.


14
22
13
25
30
11
9

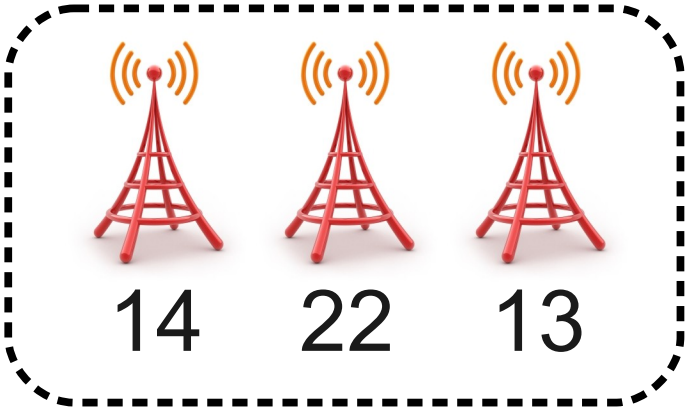

14
22
13
25
30
11
9
Maximize what's left in here.


14
22
13
25
30
11
9

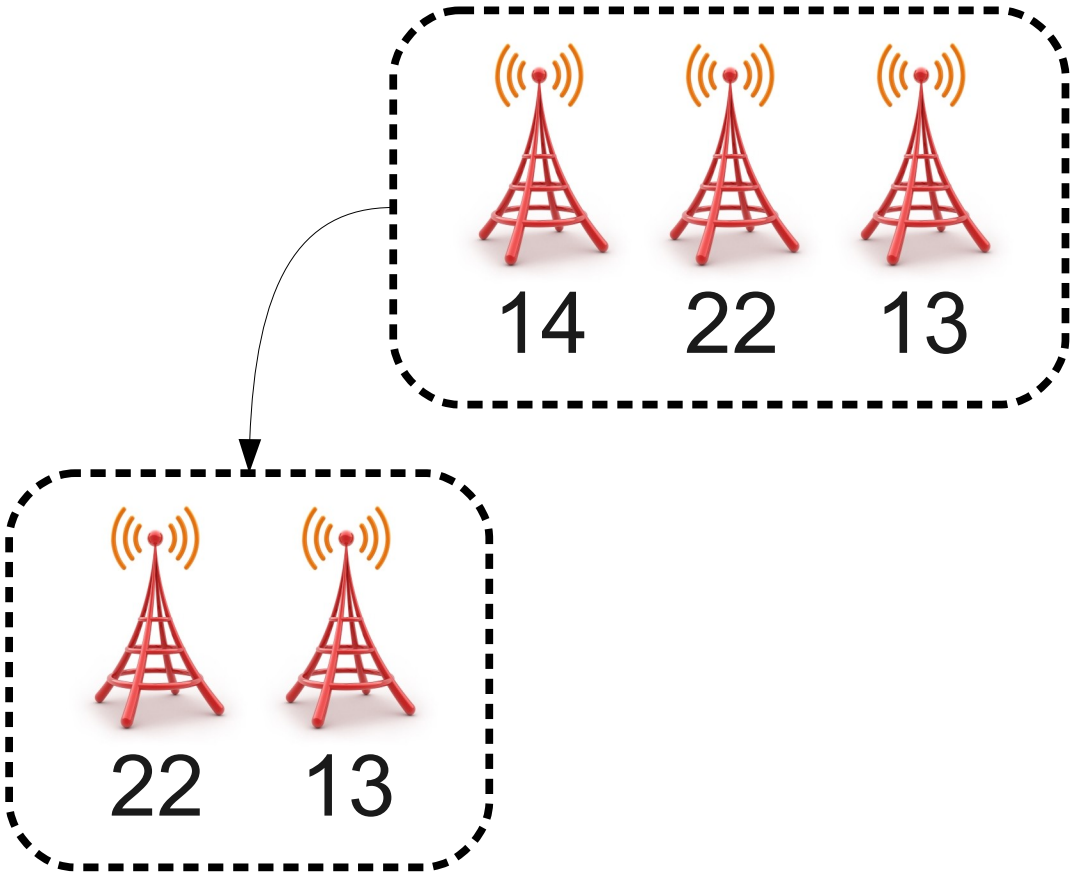

14
22
13
25
30
11
9
Maximize what's left in here.


14
22
13
25
30
11
9
Maximize what's left in here.

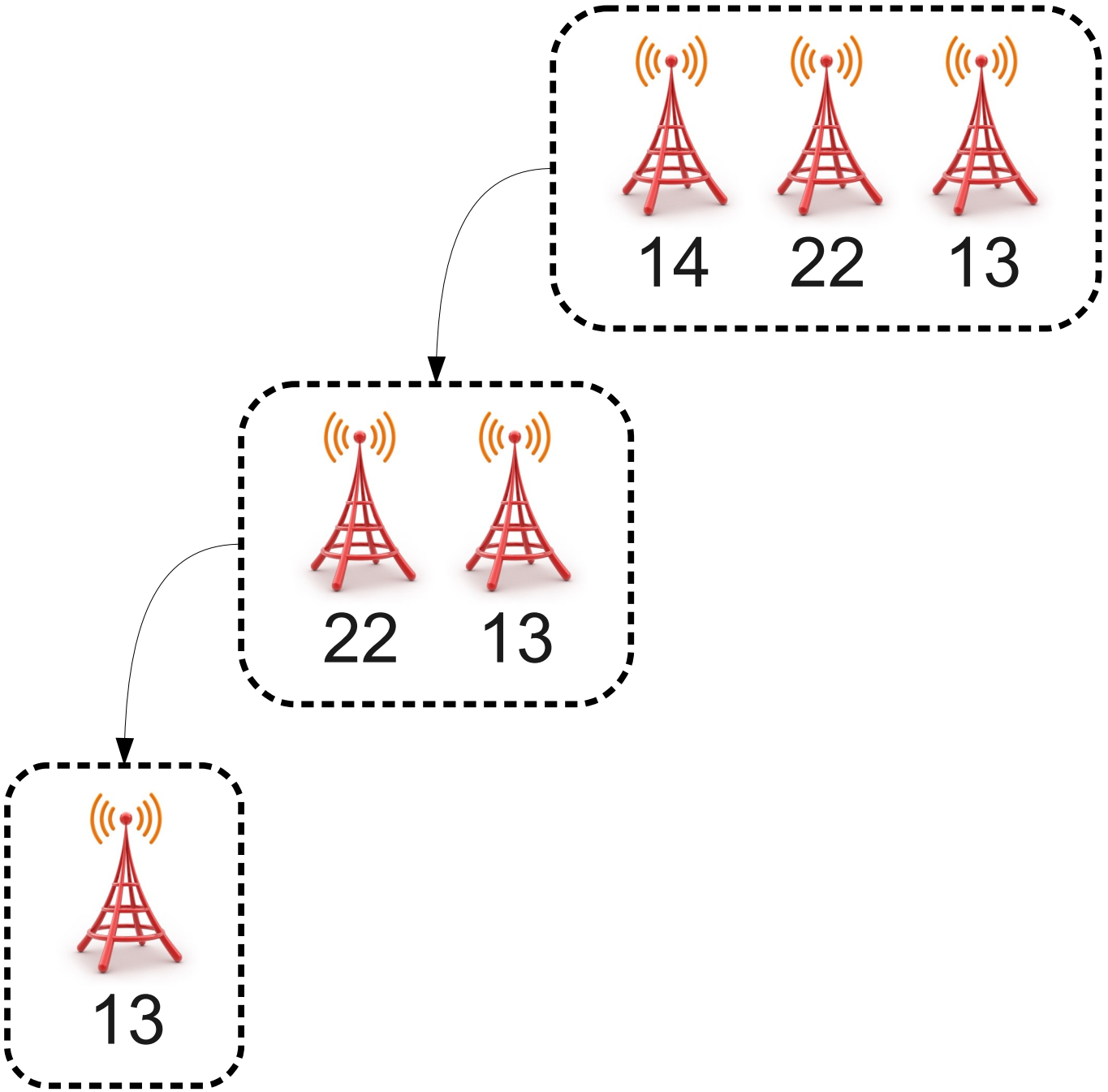
How the Recursion Works



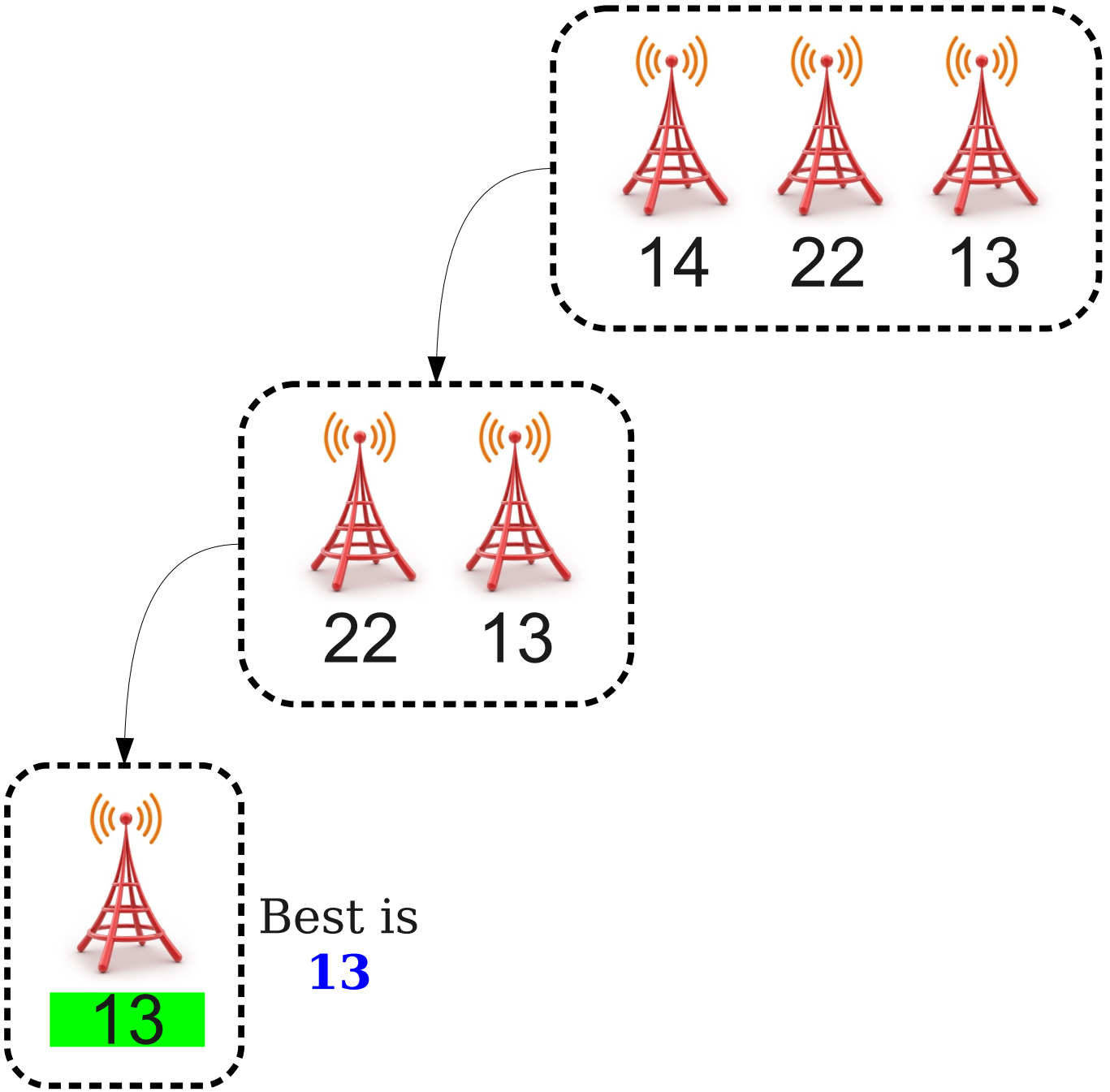
How the Recursion Works



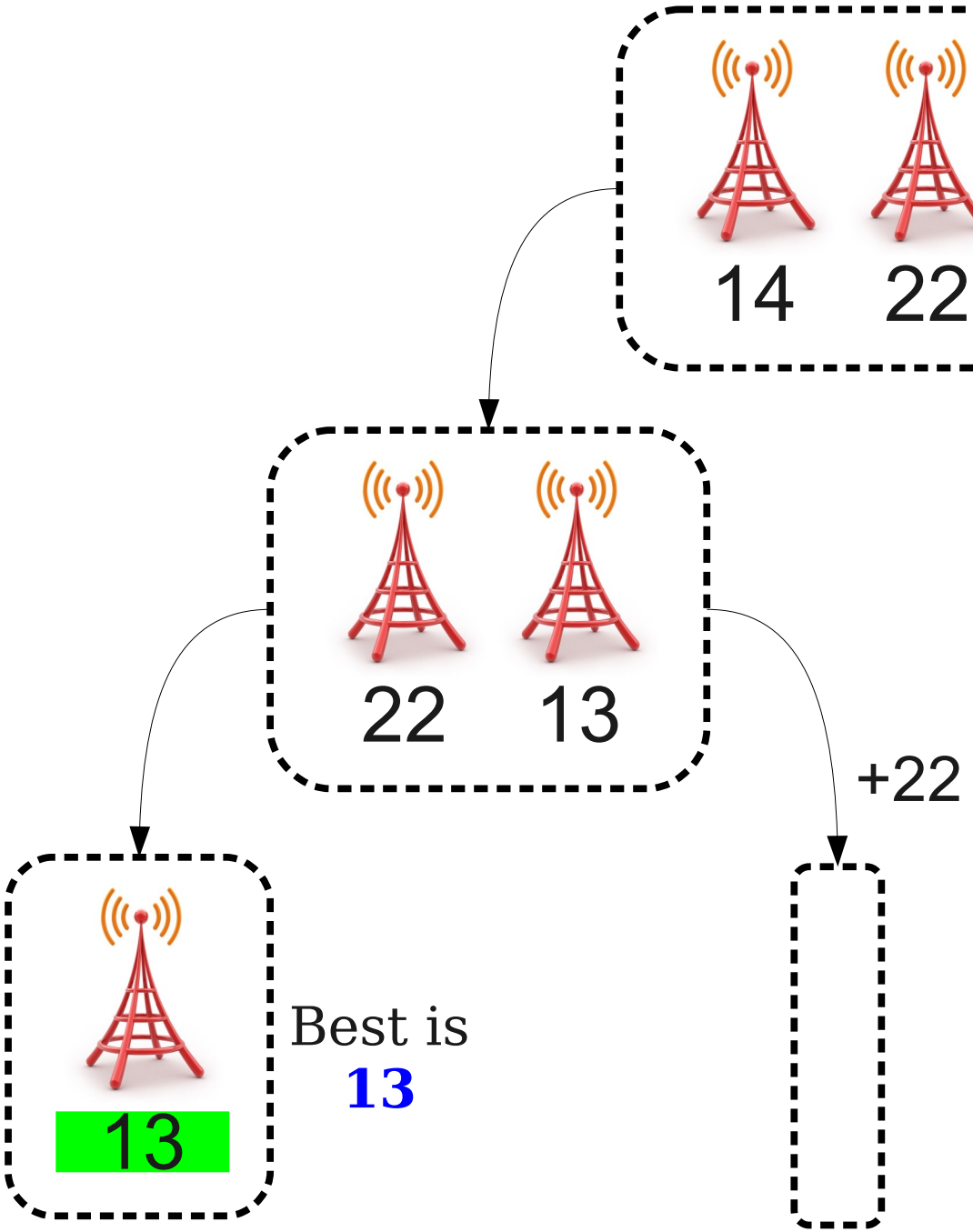
How the Recursion Works



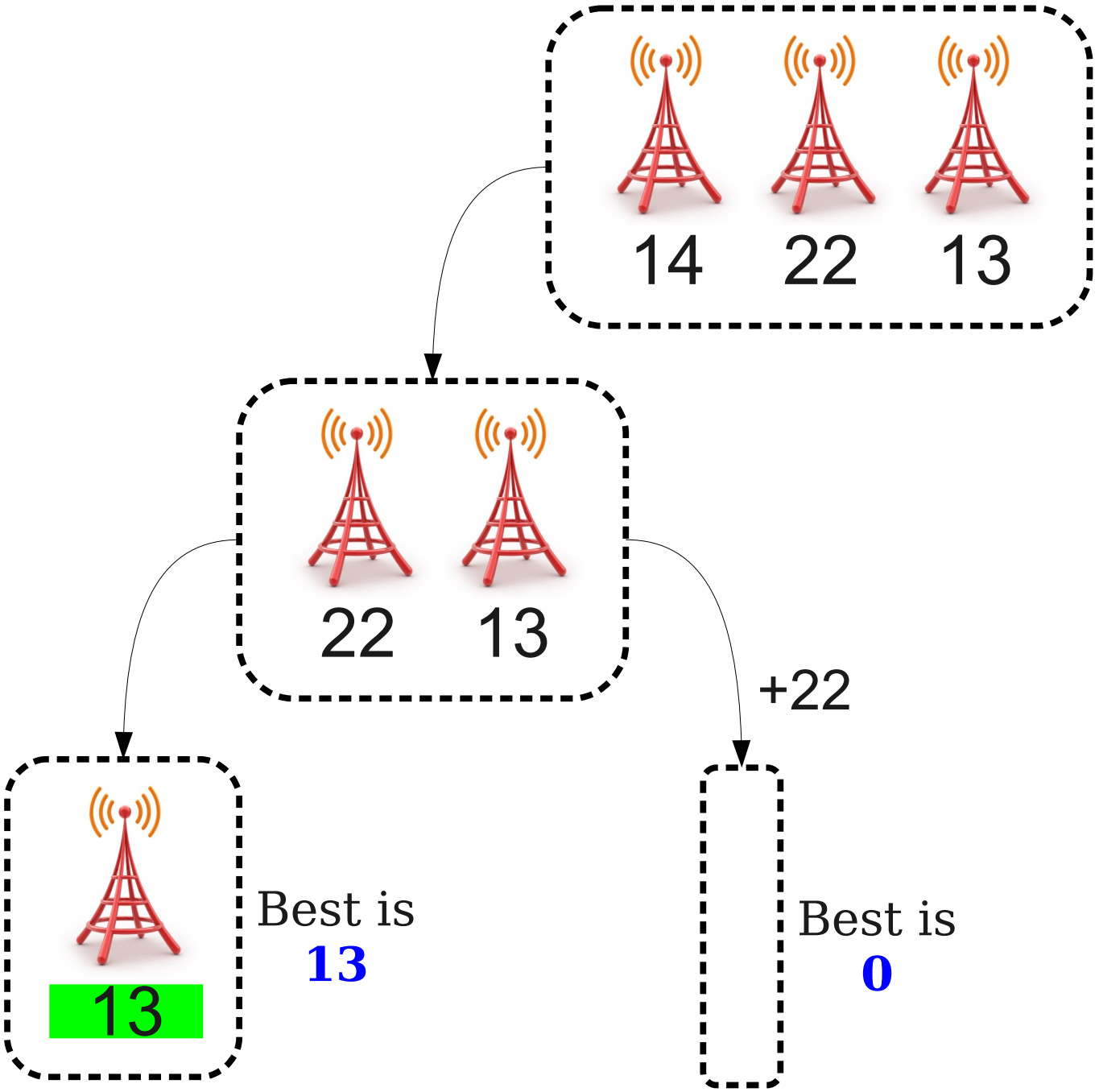
How the Recursion Works



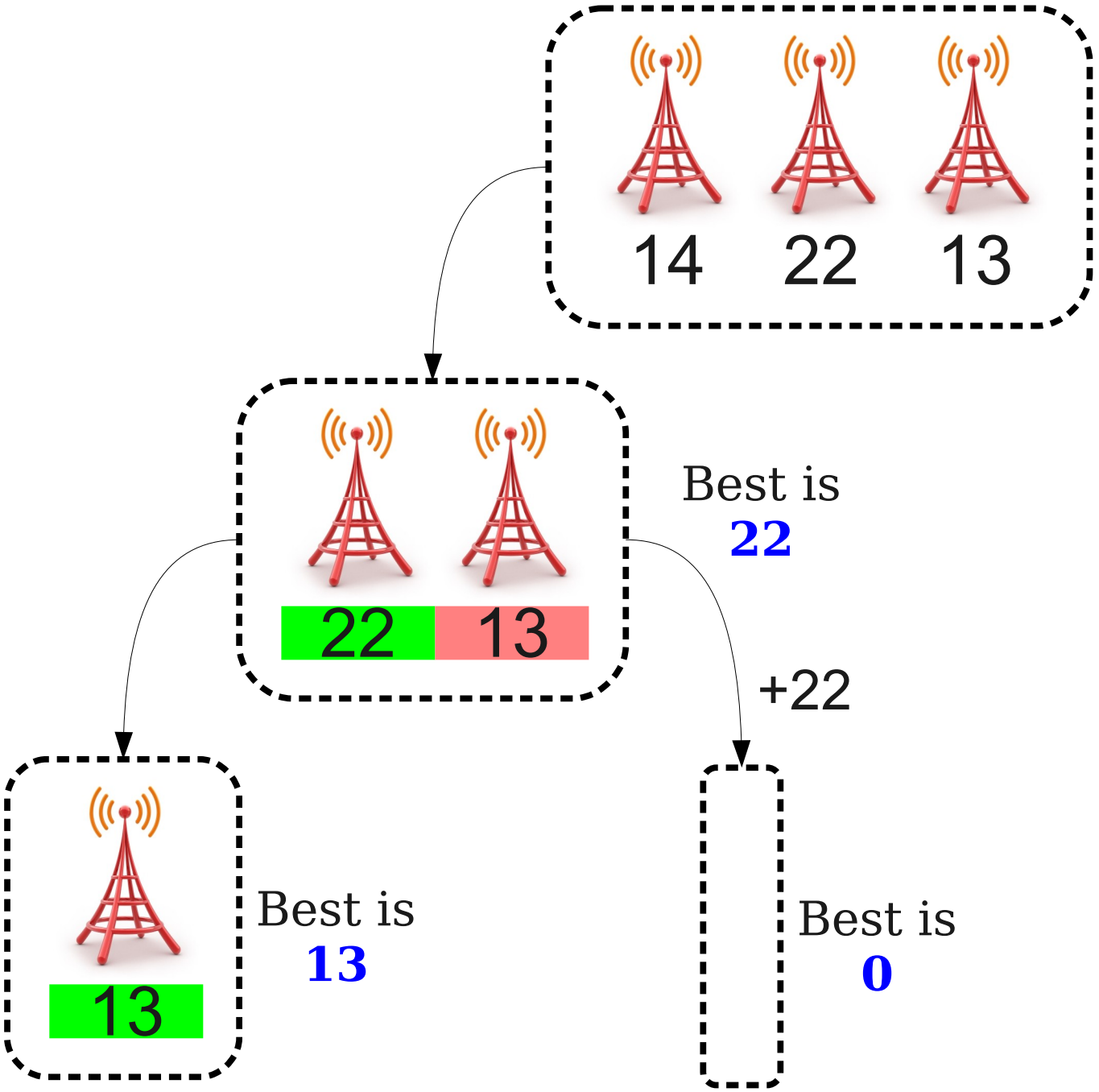
How the Recursion Works



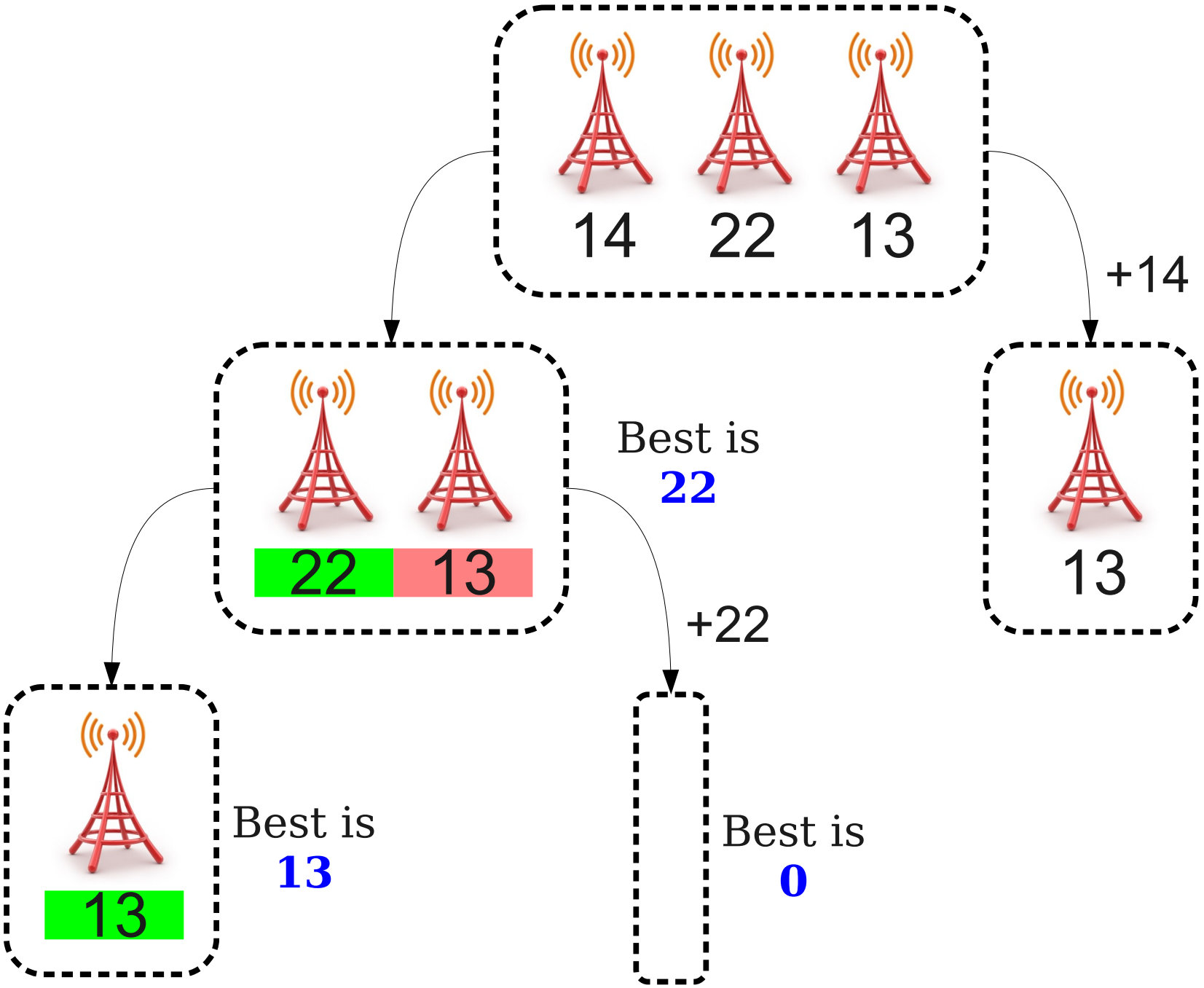
How the Recursion Works



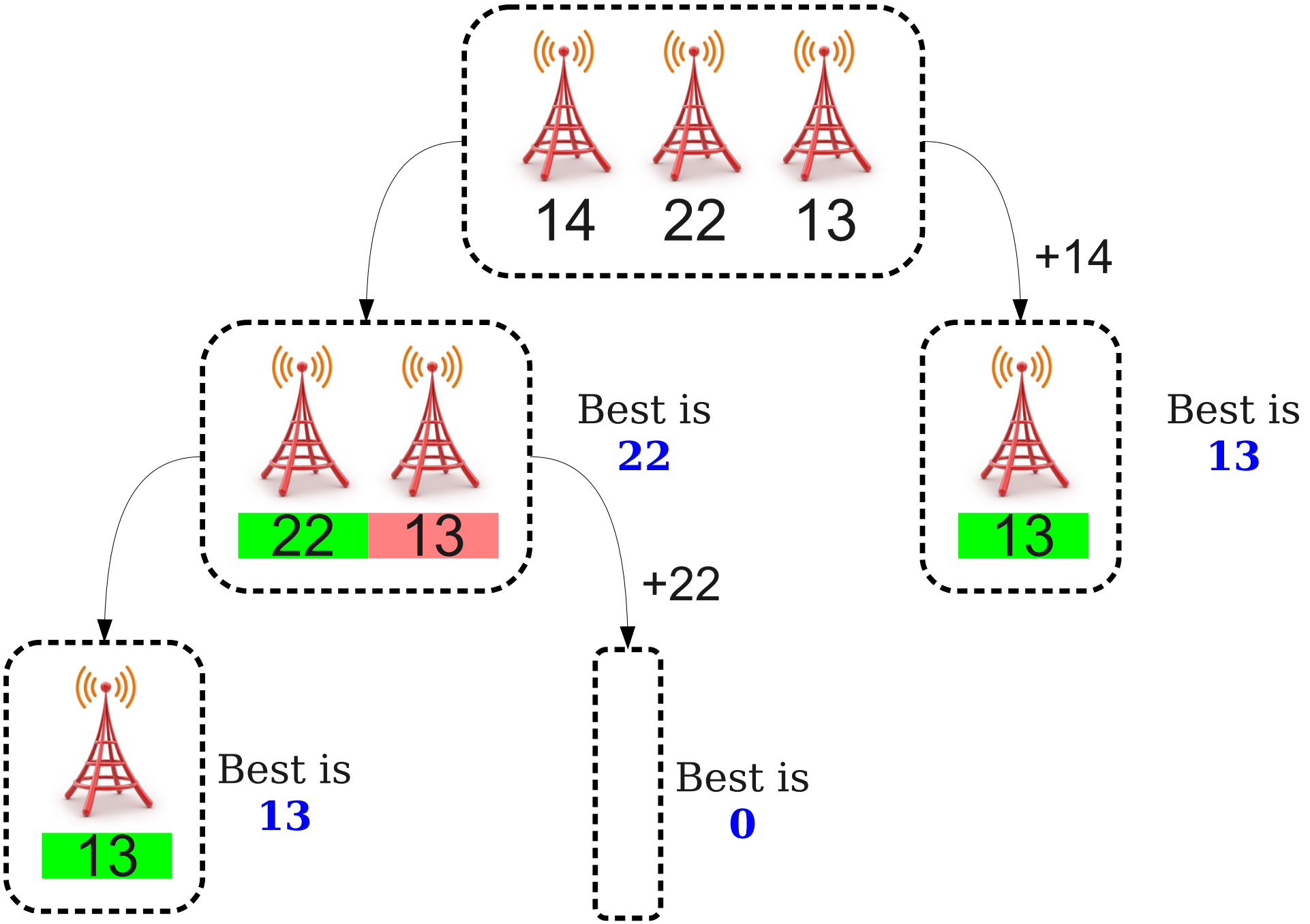
How the Recursion Works



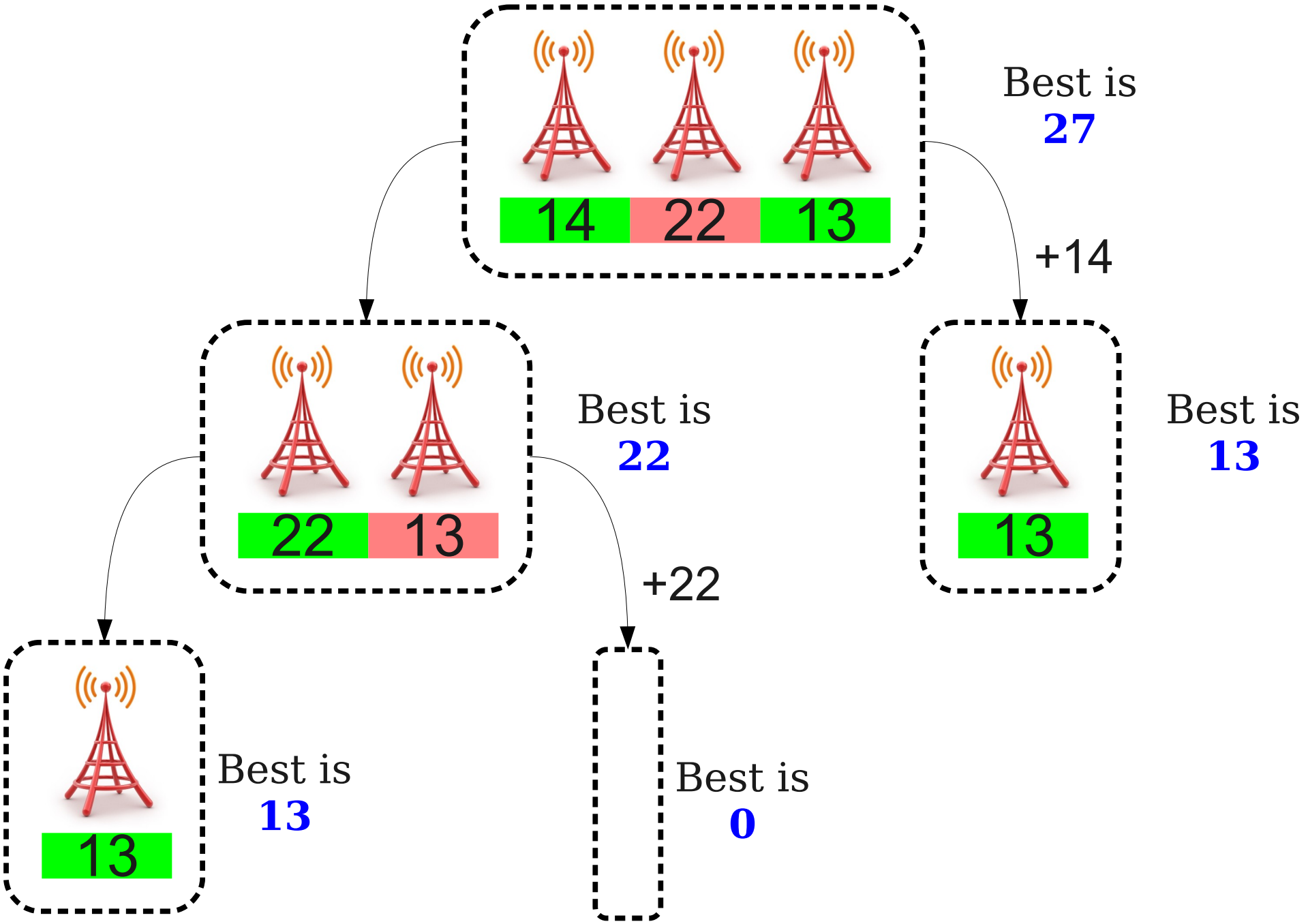
How the Recursion Works



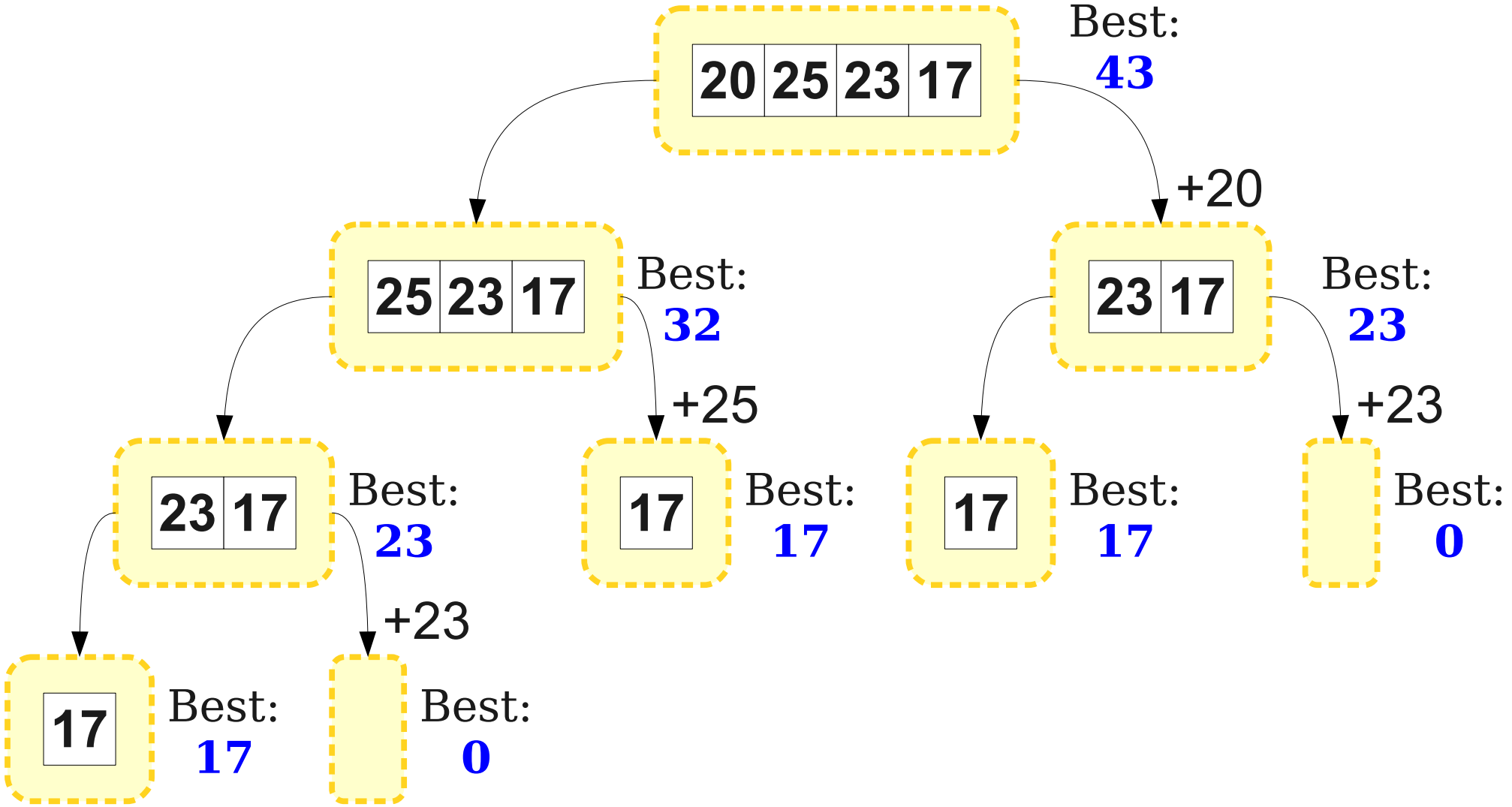
How the Recursion Works

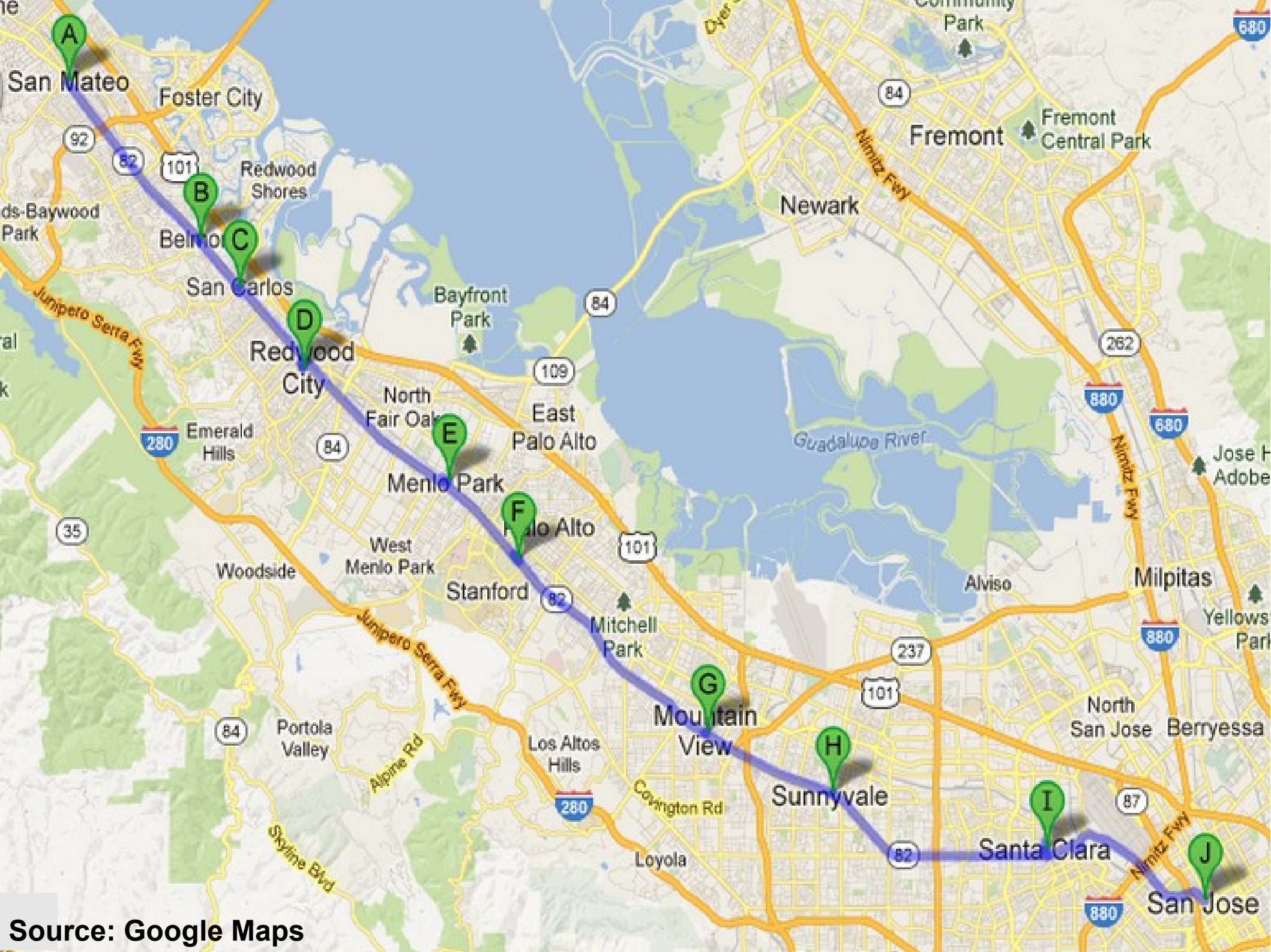


How the Recursion Works



How the Recursion Works





Source: Google Maps

Grid

Two-Dimensional Data

- The **`Grid`** type can be used to store two-dimensional data.
  - e.g. matrices, scrabble boards, etc.
- Can construct a grid of a certain size by writing

```
Grid<Type> g(numRows, numCols);
```

- Can access individual elements by writing

```
g[rows][cols]
```

Next Time

- **`Map`**
  - A collection for storing associations between elements.
- **`Set`**
  - A collection for storing an unordered group of elements.
- **`Lexicon`**
  - A special kind of **`Set`**.