Problem 1

We repeatedly add 2 records (2 * 102 bytes) and delete 1 record (100 bytes) until we have 77 records (a total of 8008 bytes used). We then add one more record (a total of 8110 bytes used) and stop because we cannot insert more records. Hence, the maximum number of records is 77+1 = 78 while the free space is 8192-8110 = 82 bytes.

Problem 2

(A)
T(R)*(40+10) = 50*T(R)

(B)
V(R, B)*(40+10) + 10*T(R) = 50*V(R,B)+10*T(R)

(C)
We need to solve the following inequality:
50*V(R,B)+10*T(R) < 50*T(R)
50*V(R,B) < 40*T(R)
V(R,B) < 4/5*T(R)

So as long as V(R,B) < 4/5*T(R) (or T(R) > 5/4*V(R,B)), indirection is more space efficient.

Problem 3

(A)
```
i=3
---------------------------------
000------>j=3
          0001
          0000
001------>j=3
          0010
          0011
010------>j=2
          <empty>
011--------^
100------>j=2
          1000
          ----
101--------^
110------>j=3
          1100
          ----
111------>j=3
          1111
          1110
```

(B)
```
i=3
---------------------------------
000------>j=3
          0001
          0000
```

```
001------>j=3
          0010
          0011
010------>j=2
          <empty>
011--------^
100------>j=1
          1000
          1111
101--------^
110--------^
111--------^
```

(C)
Minimum number of keys: = 3
Key sequence: 000000000,000000000,000000001 OR 111111111,111111111,111111110
The minimum number of buckets allocated: 2 filled + 8 empty = 10 buckets

Problem 4

(a) n_l = floor((B-P_b) / (V+P_r))

There is one block pointer pointing to the next leaf node. Also, each key and record pointer entry takes V+P_r space.

(b) n_n = floor((B-P_b) / (V+P_b))

There are n_n keys at most and n_n+1 block pointers to the nodes of the next level of the tree.

(c) Using same n: easier implementation, no changes to the algorithms
Using different n: optimize space usage

(d) Maximum number of records = (100+1)*(100+1)*100 = 1,020,100,
because first and second level have 101 pointers to next level nodes and the leaves have 100 pointers to records.

Required blocks = 1+(100+1)+(100+1)*(100+1) = 10,303,
because you have the root, 101 blocks in the second level and 101*101 in the leaf-level.

Problem 5

(a) i=2, m=11 (binary)

```
00
-----
00000
00100

01
-----
00001
00101
```

```
    10
    -----
    00010
    - - -

    11
    -----
    00011
    - - -
```

(b) i=4, m=1000 (binary)

```
    00000
    -----
    00000
    10000

    0001
    -----
    00001
    - - -

    0010
    -----
    00010
    - - -

    0011
    -----
    00011
    - - -

    0100
    -----
    00100
    - - -

    0101
    -----
    00101
    - - -

    0110
    -----
    - - -
    - - -

    0111
    -----
    - - -
    - - -

    1000
    -----
    11000
    - - -
```

Problem 6

(a) T(R) * T(S) / (V(R,A) * max(1,V(S,D)) * max(V(R,B),V(S,E))) = 1000 * 2000 / (10 * 50 * 100) = 40
(b) T(R) * T(S) / (max(V(R,A),V(S,D)) * max(V(R,B),V(S,E)) * min(V(R,A),V(S,D))) = 1000 * 2000 / (50 * 100 * 10) = 40
(c) Yes, Yes (the two equations in (a) and (b) are same)