2017-1 Computer Algorithms Homework #5

Dae-Ki Kang

May 1, 2017

(Deadline : May 8)

1. The following is the pseudo code of Heap sort. Analyze the running time of "BUILD-MAX-HEAP". Prove your analyzed running time, but the running time must be as tight as possible.

```
MAX-HEAPIFY(A, i)
l = LEFT(i)
r = RIGHT(i)
if l ≤ A.heap-size and A[l] > A[i]
    largest = l
else largest = i
if r ≤ A.heap-size and A[r] > A[largest]
    largest = r
if largest ≠ i
    exchange A[i] with A[largest]
    MAX-HEAPIFY(A, largest)
```

```
BUILD-MAX-HEAP(A)
A.heap-size = A.length
for i = ⌊A.length/2⌋ downto 1
    MAX-HEAPIFY(A, i)
```

```
HEAPSORT(A)
BUILD-MAX-HEAP(A)
for i = A.length downto 2
    exchange A[1] with A[i]
    A.heap-size = A.heap-size − 1
    MAX-HEAPIFY(A, 1)
```

2. Analyze the running time when there are m MAKE-SET, UNION and FIND-SET operations are sequentially executed where n operations of total m operations (so, obviously $n \leq m$) are MAKE-SET operations. Assume that linked list representation with weighted union heuristics is used.

   Prove your analyzed running time, but the running time must be as tight as possible. Your proof must have clear explanation.

3. Prove the followings with clear explanation:

   When you design disjoint set by forest (i.e. trees) representation with union by rank heuristics,

   - the number of nodes of which the rank k is at least 2^k.
   - if the number of all nodes is n, the expected rank of a node is $O(\lg n)$.
   - if there are m MAKE-SET, UNION and FIND-SET operations are sequentially executed where n operations of total m operations ($n \leq m$) are MAKE-SET operations, the running time of the operations is $O(m \lg n)$.

4. Write algorithms in pseudo codes with detailed explanation for the following problems:

   - Algorithm that returns a spanning tree for the give graph ($G = (V, E)$) in linear time.
   - Recall that the minimum spanning tree is a spanning tree of which the sum of edges is minimum among all possible spanning trees for the given graph.

     Now, instead of minimizing the sum of edges, present an algorithm (in pseudo code) which returns a spanning tree such that the maximum edge weight in the spanning tree is the minimum among all possible spanning trees for the given graph.
   - Suppose that all the edges in the graph is equal. Present an algorithm that returns the minimum spanning tree in the given graph.

5. Prove or disprove the above with detailed explanation.

   - PRIM and KRUSKAL work on negative weights?
   - The edge with the minimum weight in a connected undirected graph will always be included in the minimum spanning tree of the graph.
   - Suppose there is a connected undirected graph G that has a cycle C. Let the cycle C has an edge e with the maximum weight among the edges in C. Let G' be a graph after removing e. Prove that G' is still connected.
   - Continued from above question.

     Let T be a minimum spanning tree of G. Is T also a minimum spanning tree of G'?

6. Using your favorite computer programming language (but C/C++, Java, Python, C# recommended), write programs that calculate the following. The running time of the program should be less than 2 seconds.

   You and Mina is playing a game.

   Mina divides N cards into two groups. ($0 \leq N \leq 10^6$) Each card has an unique number in it.

   Mina can do one of the two things:

   - Mina randomly picks a group (let it A) and picks a card from the group A. She chooses the other group (let it B) and picks a card from the group B. She shows you the two cards.

     In the input, it will be like "`! 121 51`"
   - Or Mina randomly picks a group and picks a card from the group. Again, she randomly picks a group and pick a card from the group. Therefore the two groups can be the same or different. Only Mina knows it.

     She is asking you if the two cards are from the same group or different groups.

     In the input, it will be like "`? 121 51`"

   For Mina's question, you have to answer either "`Not yet`", "`Same`", or "`Different`"

   For fast computation, implement your program using "disjoint set forest" with "union by rank" and "path compression".

```
Input
5
? 50 100
! 50 100
? 50 100
! 100 70
? 70 50

Output
Not yet
Different
Same
```

7. Using your favorite computer programming language (but C/C++, Java, Python, C# recommended), write programs that calculate the following. The running time of the program should be less than 2 seconds.

Mina has a project to build an Internet network in YG campus with the cheapest cost possible. The buildings in YG campus are numbered from 1 to N.

Kun, the wicked owner of YG campus, has done some research and decided to have at most M connection lines between buildings where M is larger than the number of buildings N. Actually, Kun has a plan not to pay money to Mina for her work.

Mina noticed it and tries to outwit Kun by building the Internet network with the most expensive cost with the following condition:

- Sum of the connection lines is the most expensive possible.
- All the buildings are connected, which means there is a path (not necessarily a connection line) between any two buildings.
- There is no cycles in the final configuration of the Internet connection lines.

```
Input
9 14
1 2 4
1 8 8
2 3 8
2 8 11
3 4 7
3 6 4
3 9 2
4 5 9
4 6 14
5 6 10
6 7 2
7 8 1
7 9 6
8 9 7

Output
71
```

8. Using your favorite computer programming language (but C/C++, Java, Python, C# recommended), write programs that calculate the following. The running time of the program should be less than 2 seconds.

Professor Kang is going to install an Internet network on a new campus of Dongseo University. Obviously, he needs to make all buildings in the campus to be connected. In other words, any two buildings in the campus should be reached from each other through the lines of the installed network, but the two buildings don't have to be directly connected by one connection line.

When installing the network, Prof. Kang wants to minimize the cost of installing the Internet network which is the sum of constructing connection lines. Note that there are a few connection lines that were already installed for some pairs of buildings.

To help him, given a matrix to represent the possible connections of buildings and their construction costs and a list of already connected lines, your program will find the minimum amount of total construction costs so that all buildings in the campus are connected. As well as the minimum cost, your program also prints the result of the final configuration of Internet connections with the minimum cost.

The input will be a graph $G = (V, E)$. First line is N, the number of vertices. After that, an adjacency matrix of G will be given. After that, an integer K is given to denote the number of already constructed lines Finally, K lines to denote a building a is connected to a building b.

Your program has to print the minimum cost and an adjacency matrix of G will be given.

```
Input
10
 0  4  1  4  0  0  0  0  0  0
 4  0  5  0  9  9  0  7  0  0
 1  5  0  3  0  0  0  9  0  0
 4  0  3  0  0  0  0 10  0 18
 0  9  0  0  0  2  4  0  6  0
 0  9  0  0  2  0  2  8  0  0
 0  0  0  0  4  2  0  9  3  9
 0  7  9 10  0  8  9  0  0  8
 0  0  0  0  6  0  3  0  0  9
 0  0  0 18  0  0  9  8  9  0
2
0 1
0 2

Output
33
 0  4  1  0  0  0  0  0  0  0
 4  0  0  0  0  0  0  7  0  0
 1  0  0  3  0  0  0  0  0  0
 0  0  3  0  0  0  0  0  0  0
 0  0  0  0  0  2  0  0  0  0
 0  0  0  0  2  0  2  8  0  0
 0  0  0  0  0  2  0  0  3  0
 0  7  0  0  0  8  0  0  0  8
 0  0  0  0  0  0  3  0  0  0
 0  0  0  0  0  0  0  8  0  0

Input
3
  0 981 600
981   0 170
600 170   0
1
0 1

Output
170
  0 981   0
981   0 170
  0 170   0
```