EECS4020 Algorithms

Exam 2

Time: 10:10am – 12:00pm, May 11, 2022

Answer All Questions. Total = 20 + 20 + 15 + 15 + 25 + 15 = 110.

---

1. You are now in a game show, and are standing on the southern bank of a very wide river. Your challenge is to cross the river to go to the northern bank. There are n^2 boats floating on the river, and they are arranged like an $n \times n$ matrix. To cross the river, you may select any boat on the first (southern-most) row, then jump to a neighboring boat on the second row (which is either the one north of it, or northeast of it, or northwest of it), and jump to a neighboring boat on the third row, and so on, and finally reach the northern bank from any boat on the last row. Precisely, if the matrix of the boats are denoted by $B[1..n][1..n]$, the neighboring boats of $B[i][j]$ are $B[i+1][j-1]$, $B[i+1][j]$, and $B[i+1][j+1]$.

   Each boat contains a different number of gold coins, and you know exactly how many coins in each boat. If you visit a boat during the crossing of the river, you can take all the gold coins in that boat. But the time is limited! — you can only cross the river by making a "northerly" move (either north, northeast, or northwest) in each step.

   (20%) Design an $O(n^2)$-time algorithm that can compute the maximum total number of gold coins you can obtain. Explain very briefly why your algorithm is correct.

2. Consider n points located on the x-axis, with x-coordinates $x_1 < x_2 < \cdots < x_n$. We want to cover these points using unit-length line segments along the line $y = 0$. However, each line segment can cover up to two points. For instance, if we have points $1.1, 1.5, 1.8$, then we can cover these points using two unit-length line segments (say, $[1, 2]$ to cover the points 1.1 and 1.8, and $[1.5, 2.5]$ to cover the point 1.5).

   (20%) Give an $O(n)$-time algorithm to find out the minimum number of unit-length line segments we need to cover all the points. Explain clearly why your algorithm is correct.

3. In the lecture, we discussed the Inc operation to increment a binary counter. Suppose that we now have a more general kind of increment operation, called 2-Power-Inc(), where it would increment a binary counter by a value of x, for any x that is a power of 2 (e.g., x can be 1, 2, 4, 8, etc). For instance, when the content is `01101` (representing the value 13), after the 2-Power-Inc(4) operation, the content becomes `10001` (representing the value 17).

   The cost of each 2-Power-Inc operation is the number of flips used in that operation, which is the same way as we calculate the cost for the Inc operation. For instance, the operation cost of 2-Power-Inc(4) in the previous example (from 13 to 17) is 3.

   (15%) Assume that initially the binary counter has content to be all `0`s, and we perform a sequence of Inc and 2-Power-Inc operations (for the latter, each time may have a different parameter x). Assume the number of bits in the binary counter is large enough so that there will not be overflow problem during these operations. Show that the amortized cost of each operation is $O(1)$.

   Remark: Each number x that is a power of 2 has exactly one bit of `1` in its binary representation.

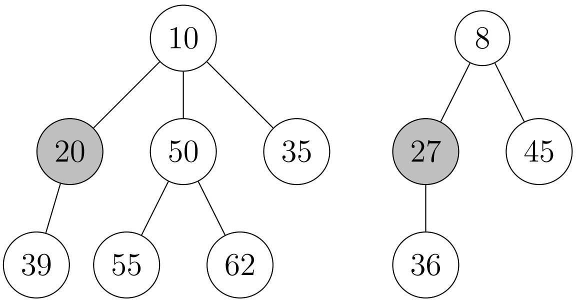
4. **(15%)** This question is about binomial heap.

   (a) Draw the structure of a binomial heap H_1 with 27 nodes.

(b) Draw the structure of a binomial heap H_2 with 7 nodes.

(c) Show the key steps when we perform union on H_1 and H_2.

5. **(25%)** Peter has maintained a set of integers using a Fibonacci heap, and the following figure shows its current status. The marked nodes are colored, and the minimum pointer (not shown) is pointing at 8.



Peter is going to perform the following sequence of operations. Describe clearly how the Fibonacci heap will change after each operation.

(a) Decrease-Key 55 to 12

(b) Extract-Min

(c) Insert 18

(d) Decrease-Key 62 to 3

(e) Extract-Min

6. There is a city called `Trellisland`. The shape of the city looks like a tree, such that each node in the tree has a building, and each edge in the tree is a road connecting two buildings. In total, there are n buildings.

Bob is an owner of a convenient store chains, and he has promised the city mayor to open his stores at the buildings, so that for each building, either the building itself, or one of its neighbors (i.e., a building connecting directly by an edge with it), will have a convenient store. However, there is a different cost to open a store at different building.

(15%) Design an $O(n)$-time algorithm to find the minimum cost for opening the stores, so that Bob can fulfill his promise.