CS 421 Algorithms (Summer 2018)

Homework #2 (75 points), Due date 6/11/2018 (Monday)

- **Q1(14 points): 0-1 Knapsack Problem:**

  There are 5 items and a knapsack. The knapsack can carry at most 6 pounds. The following table gives the values and weights of all items. If we would like to select items and put them

items	I_1	I_2	I_3	I_4	I_5
worth	13	9	5	8	17
weight	3	2	1	2	4

  into the knapsack so that the value of the load is maximized.

  (a)(10 points) Please construct and draw the necessary tables.

  (b)(4 points) Based on the tables you construct, what is the optimal solution (the items you picked).

- **Q2(16 points): Dynamic programming VS. Greedy Algorithm**

  A variant of the 0-1 knapsack problem is described as follows.
  Input: There are n items $\{1, 2, \ldots, n\}$. The i-th item weights w_i pounds, $1 \leq i \leq n$. The knapsack can carry at most W pounds.
  Output: We would like to choose items and put them into the knapsack so that the weight of the load is maximized.

  (a)(10 points) Let $m(i, j)$ denote the maximal weight of a load for a subproblem – maximizing the weight of a load by choosing items from $\{1, 2, \ldots, i\}$ and put them into a knapsack which can carry at most j pounds. Please recursively define $m(i, j)$.

$$m(i,j) = \begin{cases} \_\_\_\_\_ \\ \_\_\_\_\_ \\ \_\_\_\_\_ \end{cases}$$

  (b)(6 points) A greedy algorithm is described as follows. We first sort the items in an order of non-decreasing weights. Then, we just simply put the items, in the sorted order, to the knapsack until the knapsack cannot carry anymore items. This algorithm is incorrect. Please give a simple counter-example to disprove the algorithm.

- **Q3(10 points): Optimal Substructure Property**
  Suppose we assign n persons to n jobs. Let C_{ij} be the cost of assigning the i-th person to the j-th job. The objective of this problem is to find an assignment that minimizes the total cost of assigning all n persons to all n jobs.

  Does this problem have optimal substructure property? That is, is the following statement true? Justify your answer.
  Given an optimal assignment with the minimum cost $C = \{\ldots, C_{ij}, \ldots\}$ for the original problem (n person and n jobs), taking out the i-th person to the j-th job assignment, the remaining assignment with the cost $C' = C - \{C_{ij}\}$ must be the optimal assignment to assign $n-1$ person (excluding i-th person) to $n-1$ jobs (excluding j-th job).

- **Q4(20 points): Amortized cost analysis:**

  If we use two queues "data queue" and "working queue" to implement a stack s by the following way. Suppose that both queues have no size limit.

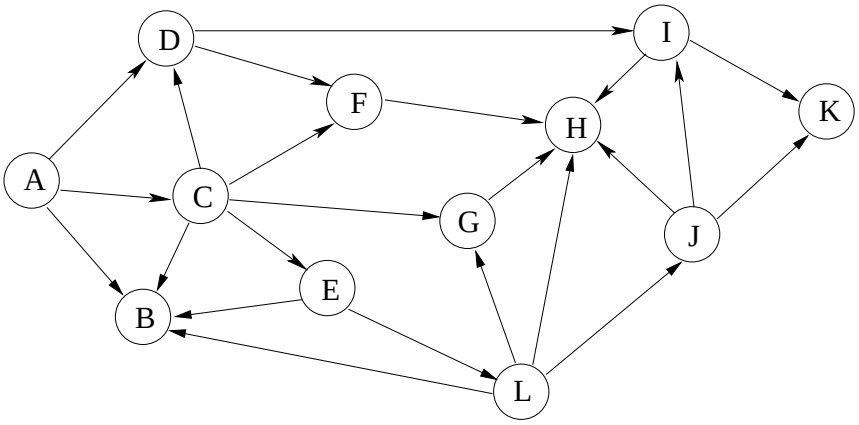
```
s.push(item)
1. enQueue the item into the data queue.

s.pop()
1. move all the items except the last one from the data queue to the working queue
2. deQueue from the data queue and return
3. now the names of the data queue and the working queue are swapped.
   That is, working queue -> data queue and data queue -> working queue
```

  (a)(10 points) Suppose that we would like to analyze the amortized costs using the accounting method. Assume there are k items in the stack before the i-th operation. If the i-th operation is a `pop` operation and we assign $2(k-1)$ amortized cost to it, then how much amortized cost for a `push` operation should be assigned? Justify your answer.

(b)(10 points) If the potential method is used, please define a potential function Φ so that the amortized costs for `push` and `pop` are constant time and linear time respectively. Assume that there are k items in the stack before the i-th operation. Justify your answer.

- **Q5(15 points): Graph Search Algorithms:**

  A weighted and directed graph is given below.



  (a)(5 points) Let vertex A be the source vertex, please find a (any) discovering sequence of vertices in a BFS search.

  (b)(5 points) Please find a (any) discovering sequence of vertices in a DFS search.

  (c)(5 points) Based on the DAG above, please find a (any) topological sequence of vertices.