CPSC 413- Fall 2020

Problem Set 2 — Algorithm Analysis

1. **(10 marks)** Exercise 1 on page 67. Justify your answers.

2. **(5 marks)** Exericse 3 on page 67. Justify your answers.

3. **(10 marks)** Prove the following statements directly from the definitions or using the Limit Test.

   (a) $2n^2 + \sqrt{n} = \Omega(n)$

   (b) $5n^3 + 3.5n^2 - 7n + 19 = O(n^3)$

   (c) $n^4 = O(2^n)$

   (d) $20n^2 + n \lg n = \Theta(n^2)$

4. **(15 marks)** Recall the Gale-Shapley Stable Matching algorithm from Chapter 1, repeated below for your convenience.

```
1  Initially all m ∈ M and w ∈ W are free
2  While there is a man m who is free and hasn't proposed to
   every woman
3     Choose such a man m
4     Let w be the highest-ranked woman in m's preference list
      to whom m has not yet proposed
5     If w is free then
6        (m, w) become engaged
7     Else w is currently engaged to m'
8        If w prefers m' to m then
9           m remains free
10       Else w prefers m to m'
11          (m, w) become engaged
12          m' becomes free
13       Endif
14    Endif
15 Endwhile
16 Return the set S of engaged pairs
```

   Section 2.3 describes more detail on the implementation of the Gale-Shapley Stable Matching algorithm, namely the data structures to use. Rewrite the Gale-Shapley algorithm to explicitely use these data structures using the following steps.

(a) Define clearly the input or arguments for the algorithm. This includes the data structures used and the type of data stored in the data structure.

(b) Show how to implement line 1 in the algorithm. How will you denote that a man or woman is free? What data structure will you use? How will you initialize this? As a result of this operation, what is the running time of this single line in the algorithm?

(c) Show how to implement line 2 in the algorithm. How will you determine if there is a free man that hasn't proposed to all women yet? What is the cost of checking this condition?

(d) Show how to implement line 3 in the algorithm. How will you choose a free man that hasn't proposed to all women? What is the cost of this choice?

(e) Show how to implement line 4 and what the cost of this implementation is.

(f) Show how to implement the condition in line 5. How will you determine if a woman is free? What is the cost of checking this condition?

(g) Show how to implement line 6. How will you keep track of engaged pairs? What data structure will be used to store these? What is the cost of adding a pair?

(h) Show how to implement line 7. How will you determine who w is engaged to? What is the cost of this operation?

(i) Show how to implement line 9. How do you indicate that m remains free? (Possibly you do nothing here.) What is the cost of this operation?

(j) Show how to implement line 11. This may the same as line 6. What is the cost of this operation?

(k) Show how to implement line 12. How does m' move from being engaged and part of a pair to being free again? What is the cost of this operation?

(l) Show how to implement line 16. What is the cost of this operation?

(m) Define the runtime of the entire algorithm as a function by:

- Adding the costs of lines 2 to 12.
- Multiplying this last result by the number of times the loop is executed.
- Adding the costs of lines 1 and 16 to this result.

(n) What is the tight (Θ) asymptotic runtime of this implementation of the Gale-Shapley Stable Matching algorithm? Prove that your bound is correct.

5. **(10 marks)** Give a worst-case analysis of the following (admittedly sub-optimal, but still useful for analysis practice!) algorithm:

- **Problem:** Create an ordered copy of an array
- **Precondition:** X is an unordered integer array $X = [x_0, x_1, \ldots, x_{n-1}]$ whose elements are of some ordered type
- **Postconditions:** Array A contains the elements of X in ascending order, X is unmodified

```
OrderedArray(X)
1   A[0] = X[0]
1   for i = 1 to n − 1
2       k = BinarySearch(A, i, X[i])
3       for j = i downto k + 1
4           A[j] = A[j − 1]
5       A[k] = X[i]
6   return A
```

For the purposes of this assignment, assume that BinarySearch(A, k, x) takes $\Theta(\lg k)$ steps, where the input array A has k elements. BinarySearch returns the index in the array here x should be inserted to maintain ascending order.

Your analysis should contain the following:

(a) An expression of the runtime function of the algorithm (should involve a sum).

(b) An asymptotic upper bound on the runtime of the algorithm. Here you must prove the upper bound. For example, to state that $2n + 3 \in O(n)$ is not sufficient, you must prove this.

(c) An asymptotic lower bound on the runtime of the algorithm. Here again you must prove your result.

(d) An asymptotic tight bound on the runtime of the algorithm.