

**Note:** Whenever explanation is requested, the explanation is worth 80% of the points unless otherwise stated.

---

1. [5 pts] Is the following statement Always True, Never True, or Sometimes True?

“ $f(n) + O(f(n)) = \Theta(g(n))$  for any non-negative functions  $f$  and  $g$  and positive  $n$ ”

Please support your answer with an explanation, i.e., a proof or an example.

2. [10 pts] Please rank the following function by increasing order of growth and explain your answer.

- $f_1(n) = n^{\sqrt{n}}$
- $f_2(n) = 2^n$
- $f_3(n) = n^{100} 2^{0.5n}$
- $f_4(n) = \sum_{i=0}^n i^3$

3. [5 pts] Please compute the tightest asymptotic bound for  $f(n, n)$ , where:

$$f(x, c) = \Theta(x) \quad \text{for } c \leq 2$$

$$f(c, y) = \Theta(y) \quad \text{for } c \leq 2$$

$$f(x, y) = \Theta(x + y) + f\left(\frac{x}{2}, \frac{y}{2}\right)$$

Note: You can use any method, as long as it is correct and the derivation of the solution is provided.

4. [20 pts] Chief Eng. Montgomery Scott (aka Scotty) discovers many communication systems from various spacecraft and satellites that have been left abandoned hundreds of years ago. After days of tinkering, he found that an essential component in each of these communication systems is an error-correcting code, which relies on multiplications of polynomials with an extremely high degree.

To test if a systems is working, Scotty would like to verify the polynomial multiplications in the system. Verifying polynomial multiplications here means that, given three polynomials,  $A(x) = \sum_{i=0}^n a_i x^i$ ,  $B(x) = \sum_{i=0}^n b_i x^i$ , and  $C(x) = \sum_{i=0}^n c_i x^i$ , he would like to know whether  $A(x) \cdot B(x) = C(x)$ . He knows that FFT can verify polynomial multiplications in  $O(n \log(n))$ .

But, since the polynomial order is often huge, Scotty would like to find a faster algorithm. He is even willing to relax the requirements of the algorithms, such that it returns the correct answer with high probability, rather than always returning the correct answer. To help Scotty develop this algorithm, please answer the following questions:

- (a) [5 pts] Describe a randomized algorithm that verifies  $A(x) \cdot B(x) = C(x)$  in  $O(n)$  time and has the following properties:
- i. If  $A(x) \cdot B(x) = C(x)$ , the algorithm outputs True.
  - ii. If  $A(x) \cdot B(x) \neq C(x)$ , the algorithm outputs True or False, and the probability that the algorithm outputs False is at least  $\frac{1}{2}$ .
- (b) [5 pts] Show that the algorithm you developed in (a) indeed satisfies property-i and property-ii above.
- (c) [7 pts] Describe a randomized algorithm to verify  $A(x) \cdot B(x) = C(x)$  that will return the correct answer with probability at least  $(1 - \epsilon)$ . Please explain how this property is satisfied by the algorithm. This explanation is worth 50% of the points. Hint: The algorithm in (a) might be useful.
- (d) [3 pts] Please derive the asymptotic upper bound of the running-time of the algorithm in (c) in terms of  $n$  and  $\frac{1}{\epsilon}$ .

5. [10 pts] Mr PQ said that he had created extremely efficient Extract-Max and Build-Heap operations for Max-Heap data structures that rely on comparison operation (i.e., the heap data structure as discussed in class). In particular, he claimed the following worst time complexity for his algorithms:

- Extract-Max runs in  $\Theta(1)$  time (Recall: Extract-Max includes retrieving an element from the heap and fixing the heap) AND
- Build-Heap runs in  $\Theta(n)$  time

Could Mr PQ's claim be True? Please explain why or why not.

6. [15 pts] Mr T would like to typeset  $n$  words neatly. Suppose each word consists of  $l_i$  ( $i \in [1, \dots, n]$ ) characters and suppose monospaced font (all characters having the same width) is used. Each line can hold at most  $C$  characters, the text is left-aligned, and the words cannot be split between lines. Mr T knows that the nicest looking paragraph would be one with the minimum extra white space at the end of each line. When a line consists of word- $i$  to word- $j$  ( $i, j \in [1, n]$ ), the number of extra white space in that line can be computed as  $E = C - (\sum_{k=i}^j l_k) - (j - i)$ . Mr T's goal is to find the typeset that will minimize the sum of extra white spaces over all lines. Please:

- [10 pts] Describe a dynamic programming algorithm for Mr T.
- [5 pts] Derive the time complexity of the algorithm you described in (a).

7. [10 pts] Suppose  $G(V, E, w)$  is an undirected weighted graph with  $V$  being the set of vertices,  $E$  being the set of edges, and  $w$  being the weight. Please answer the following questions:

- [5 pts] Suppose  $T$  is a Minimum Spanning Tree of  $G$ . Is it True that any path in  $T$  between any pairs of vertices  $v, v' \in V$  must be the shortest path in  $G$ ? Please provide a proof if it is True and a counter-example otherwise.
- [5 pts] Will the following pseudo-code returns the Minimum Spanning Tree of  $G$ ? If the answer is yes, please provide a proof. Otherwise, please provide a counter-example.

---

**Algorithm 1** Algorithm-1( $G(V, E, w)$ )

---

```
1:  $T = \emptyset$ 
2: for each edge  $e \in E$ , taken in arbitrary order do
3:   if  $T \cup e$  has no cycles then
4:      $T = T \cup e$ 
5: Return  $T$ 
```

---

8. [5 pts] Consider a hash table  $T$  with  $m$  slots. Suppose  $T$  contains a single element, and this element has key  $k$ . Ms Search has been searching for  $r$  keys that are different from  $k$  in hash table  $T$ . Assuming  $T$  uses simple uniform hashing, is  $\frac{r}{m}$  the probability that at least one of the  $r$  searches collide with the slot containing key  $k$ ? If this probability is correct, please provide a derivation to support it. Otherwise, please provide the new probability and its derivation.

**oOo That's All Folks oOo**