COMPSCI 320SC 2016 Test

Attempt *all three* questions. (Use of calculators is NOT permitted.)

Put the answers in the space below the questions. Write clearly and *show all your work*!

Marks for each question are shown below and just before each answer area.

This 50 minute test is worth 10% of your final grade for the course.

Question #:	1	2	3	Total
Possible marks:	10	10	10	30
Awarded marks:				

University ID: _____

Student Name: _____

Student Signature: _____

Time Finished: _____

1. (a) Who are the authors of the CompSci 320 textbook: *Algorithm Design*?

(***2 marks***)

(b) Explain the *Stable Matching Problem* that was discussed in the first week of CompSci 320.

(***2 marks***)

(c) Name four algorithm design techniques (paradigms) that we will cover in CompSci 320?

(***2 marks***)

(d) Name three of the top (most influential) algorithms as mentioned in CompSci 320 lectures?

(***2 marks***)

(e) What are the names of the Lecturer and Tutor for CompSci 320?
Bonus (+1 mark): give the name of the Marker.

(***2 marks***)

2. Consider the following divide-and-conquer "algorithm":

```
function printer(int n)
    for i = 1 to n^2 do
        j = 1
        while  j ≤ √i do
            printline "hello world"
            j = j + 1
    if n > 0 then
        for i = 1 to 4 do
            printer(⌊n/2⌋)
```

Let $T(n)$ denote the number of lines of output generated by a call of printer(n). For the following, you may assume $\sum_{i=1}^{n} \sqrt{i} = \Theta\left(n^{\frac{3}{2}}\right)$.

(a) What is the value of $T(1)$, $T(2)$, and $T(4)$? (***3 marks***)

(b) Provide a recurrence equation for $T(n)$. (***4 marks***)

(c) Solve the recurrence asymptotically for general n. (***3 marks***)

3. We need to merge a set $S = \{f_1, f_2, \ldots, f_n\}$ of sorted files of different lengths using an optimal *merging pattern* where the merging of two files f_i and f_j costs the sum of their lengths $|f_i| + |f_j|$. The total merging cost $C(T)$ of a merging pattern tree T is analogous to the Average Bits per Letter (ABL) cost of a prefix code ("Huffman Tree"). Here we sum the internal files/nodes (repeatedly merging) implicitly via the files' depths in the merging pattern:

$$C(T) = \sum_{k=1}^{n} |f_k| \cdot \text{depth}(f_k)$$

A greedy algorithm that finds an optimal merging pattern is as follows:

agorithm Merging_Pattern($S = \{f_1, f_2, \ldots, f_n\}$)
new PriorityQueue P
for $i = 1$ **to** n **do**
Store f_i in P indexed by its length $|f_i|$
while P is not empty **do**
(a) Extract two smallest elements f_i and f_j from P
(b) Merge f_i and f_j and insert new file (parent node $f_{p=i+j}$) in P indexed by $|f_i| + |f_j|$

(a) What is the running time of this algorithm? Explain any implementation issues and indicate how to easily compute $C(T)$ from this algorithm. (***3 marks***)

(b) For files of lengths $|f_1| = 1, |f_2| = 2, |f_3| = 4, |f_4| = 4, |f_5| = 7, |f_6| = 9, |f_7| = 20, |f_8| = 25$ draw an optimal merging pattern as a rooted tree T and compute its cost $C(T)$. (***3 marks***)

(c) Show the correctness of this algorithm by giving a formal proof of this claim: For a set S there exists an optimal merging pattern such that the two shortest files f_i and f_j from S are first merged together. (***4 marks***)

Scratch Page—will not be marked