CSCE5150
Analysis of Computer Algorithms

Dr. Xuan Guo

1

Content

- Midterm Exam
- Topics
- Sample questions

2

Midterm Exam

- Friday, March 21ST
- 10:00 a.m. - 12:00 p.m., 120 minutes
- Paper-based exam, 4 problems with 1 bonus problem
- Open-book/note
- Sharing materials with classmates is **NOT** allowed. **NO** use cellphones, laptops, or access online resources. Any violation will be considered cheating, resulting in a score of 0.

3

Topics

- Asymptotic notation
  - Θ, O, Ω
  - Running time analysis
- Divide-and-Conquer
- Greedy algorithms
- Dynamic programming
- Flow network

4

Problem types

- True/False
- Short answers
- Algorithm design

5

Sample question 1

- What is the definition of Big-O?

6

Sample question 2

- If the divide-and-conquer convex hull algorithm used a $\Theta(n^2)$ strategy to discover the maximum and minimum tangents, the overall algorithm would run in $\Theta(n^2 \log n)$ time.

- MSTs can be used to find the minimum cost path between any two nodes.

7

Problem: Convex Hull
Divide and Conquer - Analysis

$$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n)$$
$$= \Theta(n \log n)$$

1. Divide into left half A and right half B by x coordinates
2. Compute $CH(A)$ and $CH(B)$, recusively
3. Combine CH's of two halves (merge step)

    1. Find upper tangent (a_k, b_l)
    2. Find lower tangent (a_u, b_v)
    3. Cut and paste in time $\Theta(n)$.

```
i=1
j=1
while (y(i, j + 1) > y(i, j) or y(i − 1, j) > y(i, j))
     if (y(i, j + 1) > y(i, j)) [ move right clockwise]
               j = j + 1( mod q)
     else
               i = i − 1( mod p) [ move left anti-clockwise ]
return (ai, bj) as upper tangent
```

8

8

Sample question 3

Minimum number of guesses needed to find an array size.

Given a sorted array, in worst-case, what is the minimum number of guesses needed to find the array size. You can only specify the index to acquire the corresponding element in this array at a time. You will get NULL if the index is out of range.

10

Sample question 4

Activity Selection Problem

You are given n activities with their start and finish times. Select the maximum number of activities that can be performed by a single person, assuming that a person can only work on a single activity at a time.

11

Sample question 4 cont.

The greedy choice is to always pick the next activity whose finish time is the least among the remaining activities and the start time is more than or equal to the finish time of the previously selected activity.

We can sort the activities according to their finishing time so that we always consider the next activity as the minimum finishing time activity.

1. Sort the activities according to their finishing time
2. Select the first activity from the sorted array and print it
3. Do the following for the remaining activities in the sorted array
4. If the start time of this activity is greater than or equal to the finish time of the previously selected activity, then select this activity and print it

12

Sample question 4 cont.

Proof:

Let A be the greedy choice (starting with 1st activity in the sorted array). Let there be another choice B starting with some activity k ($k \neq 1$, so $\text{finishTime}(k) >= \text{finishTime}(1)$) which alone gives the optimal solution.

So, B does not have the 1st activity and we can make B more likes A without changing the number of activities by:

$$A \leftarrow \{B - \{k\}\} \cup \{1\}$$

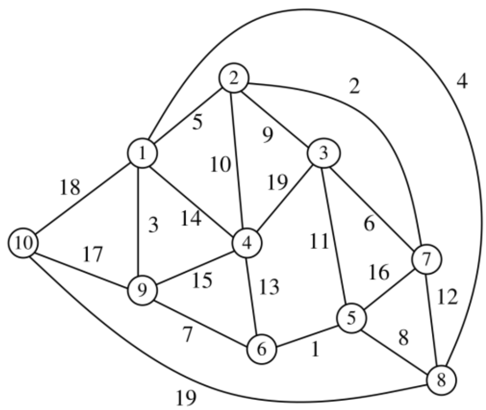
Repeat until B turning into A.

We conclude that $|A| = |B|$, therefore activity A also gives the optimal solution.

13

Sample question 5

Run Prim's algorithm on the following graph starting vertex 1. List the edges in the order in which they are chosen (you may use the edge costs for this purpose).



14