

**CPSC 510 Algorithms
Winter 2012**

Midterm Exam Practice Problems (Solution)

1.

a. $T(n) = \Theta(2^n \cdot n^3)$

b. $T(n) = \Theta(n^2 \lg n)$

2.

FIND-FIRST-UPPER($S, start, end$)

```
1  if  $start == end$ 
2    if IS-UPPER( $S[start]$ )
3      return  $start$ 
4  else
5    return  $start + 1$ 
6  let  $mid = \lfloor (start - end) / 2 \rfloor$ 
7  if IS-UPPER( $S[mid]$ )
8    FIND-FIRST-UPPER( $S, start, mid - 1$ )
9  else
10  FIND-FIRST-UPPER( $S, mid + 1, end$ )
```

Divide and conquer recurrence: $T(n) = T(n/2) + \Theta(1)$

$$T(n) = \Theta(\lg n)$$

3.

LONG-PATH(G)

```
1  let  $n$  be the number of vertices of  $G$ 
2  let  $M$  be an array of integers from  $[1$  to  $n]$ 
3   $M[1] = 0$ 
4  for  $i = 2$  to  $n$ 
5     $max = \text{NO-PATH}$ 
6    for each edge  $(v_j, v_i)$ 
7      if  $M[j] \neq \text{NO-PATH}$  and  $M[j] + 1 > max$ 
8         $max = M[j] + 1$ 
9     $M[i] = max$ 
10 return  $M[n]$ 
```

4. The greedy choice is to select the participant that will take the longest to complete the biking and running stages ($b_i + r_i$)

PRODUCE-SCHEDULE(P)

1 sort P by descending $b_i + r_i$

Proof of how the greedy choice leads to an optimal solution:

Let O be an optimal solution that does not use any order that could be produced by the greedy algorithm. This means that the optimal solution must contain two participants i and j such that j is sent out directly after i , but $b_i + r_i < b_j + r_j$. We'll assume that participant i is sent out at time T . This means that participant j will finish later than participant i at time $T' = T + s_i + s_j + b_j + r_j$. Now, swap the order of these participants, such that j swims before i . In this situation, both participants i and j will finish before time T' : Participant j no longer has to wait for participant i to swim and finishes at time $T + s_j + b_j + r_j < T'$. Participant i now finishes at time $T + s_i + s_j + b_i + r_i$ but since $b_i + r_i < b_j + r_j$, that finishing time is earlier than T' . This proves that swapping the order of these participants does not increase the overall completion time. By swapping all inversions like this, you arrive at the same solution as the greedy algorithm. This proves the greedy algorithm is optimal.