

# Homework 3

CS 4050 / 7050

## Greedy Algorithms

### Part A

Suppose you are given two sets  $A$  and  $B$ , each containing  $n$  positive integers. You can choose to reorder each set however you like. After reordering, let  $a_i$  be the  $i^{th}$  element of set  $A$ , and let  $b_i$  be the  $i^{th}$  element of set  $B$ . Your goal is to maximize the product

$$\prod_{i=1}^n a_i^{b_i}$$

Fill in the cell below with an enumerated greedy algorithm. Ensure it renders properly when you export the notebook to PDF for grading.

Double-click this cell and add your algorithm below.

### MaxProduct(A,B)

1. A= selectionSort(a)
  2. B= selectionSort(b)
  3. total = 0
  4. for(i from 0 to n)
  5.     total \*= a[i]^(b[i])
  6. return total
- 

### Part B

Prove that your algorithm finds the optimal maximum product.

Double-click this cell and add your answer below.

By sorting both A and B in order from largest to smallest this ensures a maximum product because the largest numbers in A are set to the power of the largest numbers in B. For example,  $2^{32}$  is much smaller than  $9^{32}$ , so largest numbers in A should be matched up with largest numbers in B for a maximum product. For any integer and exponent combination, the larger they both are the larger the result is. This could

## Part C

Consider a very large array of sorted numbers. Imagine that you have a query number that you must use to rank all the numbers in the sorted array based on distance to the query,  $q$ .

That is, for each  $Arr[i]$ , you must measure  $Dist[i] = |q - Arr[i]|$  and then  $sort(Dist)$  and report top  $k$  matches. In essence, the 1-dimensional version of the Programming/Analysis assignments.

Write a greedy algorithm that will find the top  $k$  in  $T(n) = \log_2(n) + 2k$

Double-click this cell and add your answer below.

```

binarysearch(Arr[i] for query q)
while(total < k)
    measure distance of left and right to i, store distances as L and R
    if(l<r) add l
        total ++
        l = left of l
    elseif(r<l) add r
        total++
        r = right of r
    else
        add l and r
        total +=2
        l = left of l
        r = right of r

```

## Part D (Optional, no points to be lost or gained)

Contemplate Programming/Analysis assignments 1 and 2. In PA1, we wrote algorithms that were brute force, and ran in approximately  $T(n) = 13n + n\log_2(n) + k$ .

Then, in PA2, we rewrote the solution to achieve (usually)  $T(n) = 13n + k\log_2(k) + k$ .

Can your algorithm above be extended into the 12-dimensional space of the PA assignments?

Can you achieve a sub-linear time, e.g.  $T(n)$  strictly  $< O(N)$ ?

Double-click this cell and add your answer below.

---

## Minimum Spanning Trees

### Part E

Given a graph  $G$  and a MST  $T$ , suppose that we decrease the weight of one of the edges not in  $T$ .

Give an algorithm for finding the MST in the modified graph.

Double-click this cell and add your answer below.

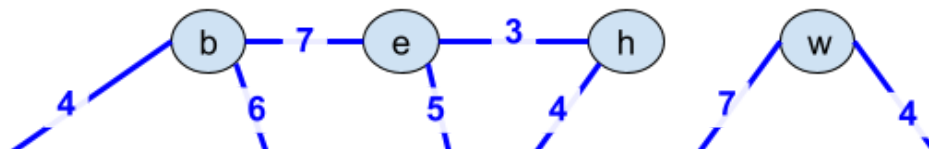
```
add edge with new weight to minimum spanning tree, creating a loop
for edges in loop, find heaviest
remove heaviest edge in loop
```

---

### Part F

Given the graph below, list the MST based on Dijkstra's Algorithm and state total traversal cost from node  $a$ .

```
![WeightedGraph]
(https://web.dsa.missouri.edu/static/images/WeightedGraph\_MST.png)
```



Double-click this cell and add your answer below. If you want to add an image, you can upload it to this folder after drawing and converting to JPG, GIF, PNG in another application. [See: How to Insert Embedded Image \(https://stackoverflow.com/questions/10628262/inserting-image-into-jpython-notebook-markdown\)](https://stackoverflow.com/questions/10628262/inserting-image-into-jpython-notebook-markdown)

```

a,b,f,c,e,d,g,h,i,k,j,m,q,y,w,n,x,z
a-0, b-4, c-10, d-12, e-11, f-10, g-13, h-14, i-16, j-20, k-17, m-22,
n-31, q-23, w-30, x-32, y-28, z-34
shortest path from a-z: 34
total weight of the path traveled: 86

```

**The End**