

## Assignment 2

### Question 1. [10 MARKS]

In the following let  $A = \{a_1, \dots, a_n\}$  be a set of  $n$  items where item  $a_i$  has value  $v_i$  and capacity  $c_i$ . Suppose you have a bag with maximum capacity  $C$ . We will develop a recurrence relation to find the maximum value obtained by a subset of  $A$  whose capacity does not exceed  $C$ . *Note: this is the classic **Knapsack Problem**. Many solutions on-line use linear programming or dynamic programming, but this is **not** what we are asking you to do here; you do not need anything other than the material you learned in class to answer this question.*

#### Part (1) [2 MARKS]

Let  $T(i, c)$  be the maximum value obtained after considering the first  $i$  items with total capacity  $c$ . Come up with an equation for  $T(0, c)$  and  $T(1, c)$  i.e. what is the maximum value obtained if we considered none of the items? Only consider the first item?

#### Part (2) [3 MARKS]

Suppose we know  $T(i, c)$ , the maximum value obtained after considering the first  $i$  items with total capacity  $c$ . Find an equation for  $T(i + 1, c)$  in-terms of the values  $T(i, 0), T(i, 1), \dots, T(i, c)$ . Use this equation along with part one to come up with a general recurrence relation for  $T(k, c)$ . How would we find the maximum value obtained by a subset of  $A$  whose capacity does not exceed  $C$ ?

#### Part (3) [5 MARKS]

Suppose we have two bags with capacities  $C_1$  and  $C_2$  respectively. Let the total profit be the sum of the values of all items in both bags. Find the maximum total profit subject to the capacity constraints as we did for the one bag case above.

### Question 2. [11 MARKS]

In the following, consider an  $m \times n$  grid as shown in Figure 1. The bottom left point and top right points are  $p_{1,1}$  and  $p_{m,n}$  respectively. All other points are indexed in the standard way.

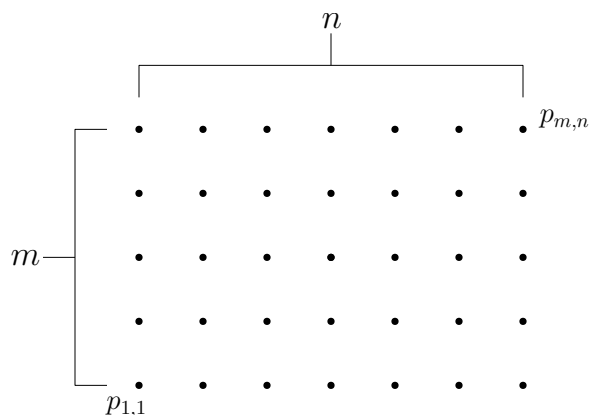


Figure 1: The grid of points we will consider for this problem.

**Part (1)** [6 MARKS]

Find a recurrence relation for the number of axis aligned rectangles in an  $m \times n$  grid. See Figure 2 for an example where  $m = 3$  and  $n = 3$ . Explain your process.

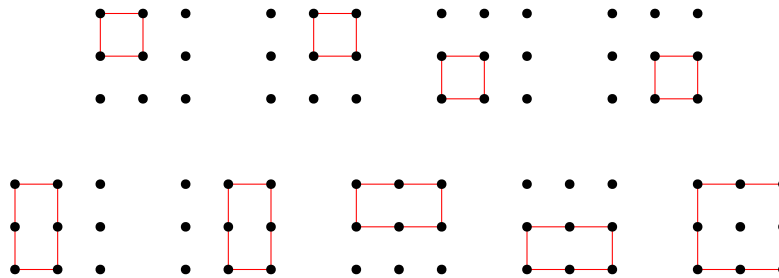


Figure 2: Example with  $m = n = 3$ . There are nine possible axis aligned rectangles possible.

**Part (2)** [1 MARK]

Hypothesize a closed form for your recurrence relation. *Hint: check to see if your closed form is correct in the special case when  $m = n$  using the on-line encyclopedia of integer sequences.*

**Part (3)** [4 MARKS]

Prove your closed form using induction.

**Question 3.** [10 MARKS]

An array is said to be **rotated** by one space if all its elements are moved ahead circularly, that is every element is moved to the next position and the last element is moved to the first position. For instance the array  $[1, 2, 3, 4, 5]$  rotated by one space gives  $[5, 1, 2, 3, 4]$ . Rotating an array by  $r$  spaces simply means rotating it by one space  $r$  times. So the array  $[1, 2, 3, 4, 5]$  rotated 2 spaces will give  $[4, 5, 1, 2, 3]$  and rotated 5 spaces will give  $[1, 2, 3, 4, 5]$  back again.

The input is an array  $A$  of  $n$  **distinct** elements which is sorted in ascending order and then rotated a certain number of spaces. (The number of spaces the array is rotated may be zero as well.) You have to devise an algorithm **minimum**, which finds the minimum element of the array.

**Part (1)** [2 MARKS]

Devise an  $\mathcal{O}(n)$  brute-force algorithm for **minimum** in Python notation. Give an informal explanation of why the time complexity is  $\mathcal{O}(n)$ .

**Part (2)** [3 MARKS]

Devise a divide-and-conquer algorithm for **minimum** which performs better than the brute-force version. (Be careful of the edge and base cases.)

**Part (3)** [1 MARK]

Give recurrence relation  $T(n)$  for the worst-case time complexity for the divide-and-conquer algorithm.

**Part (4)** [4 MARKS]

Use repeated substitution to get the closed form for  $T(n)$ . Prove using induction that the closed form indeed satisfies the recurrence relation.

**Question 4.** [9 MARKS]

Apply the Master Theorem to derive an asymptotic upper bound for each of the following recurrences.

**Part (1)** [3 MARKS]

$$T(n) = \begin{cases} 4 & n = 1 \\ 4 \cdot T(\lfloor n/2 \rfloor) + n^2 + 8n & n > 1 \end{cases}$$

**Part (2)** [3 MARKS]

$$T(n) = \begin{cases} 1 & n = 1 \\ 2 \cdot T(\lfloor n/4 \rfloor) + 1 & n > 1 \end{cases}$$

**Part (3)** [3 MARKS]

$$T(n) = \begin{cases} 10 & n = 1 \\ 9 \cdot T(\lfloor n/3 \rfloor) + n^3 \cdot \log n & n > 1 \end{cases}$$