Algorithms

Assignment: Dynamic Programming

Name: ....

Id: ....

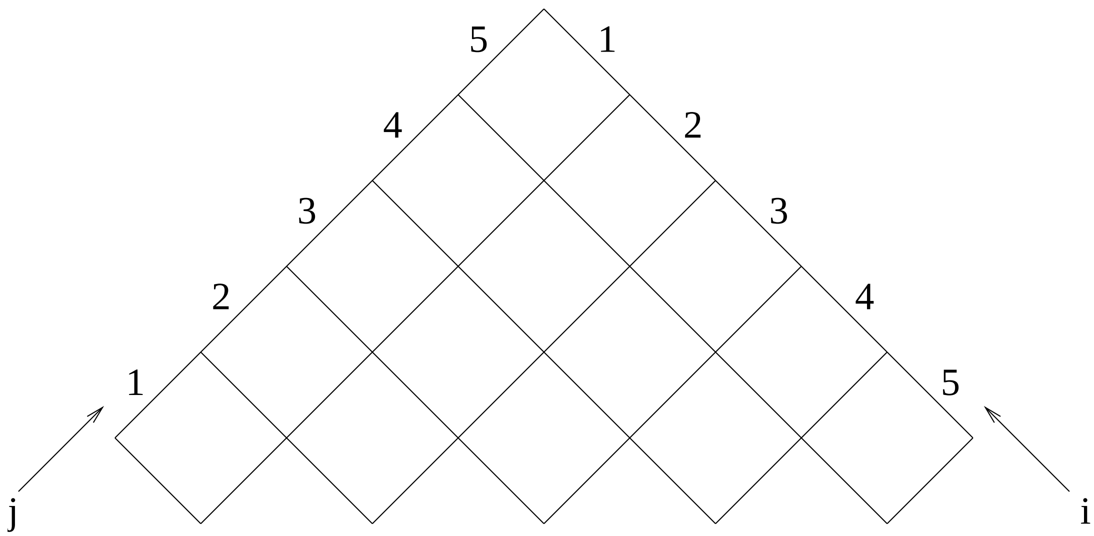
Grade

Good Luck!

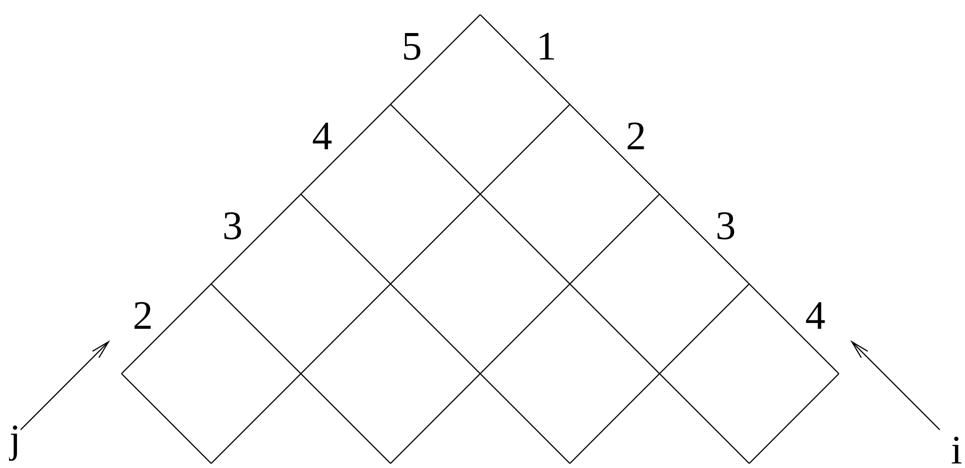
1. Run the Matrix-Chain Multiplication algorithm with the following input:

$$A_1[13, 21] \;\; A_2[21, 5] \;\; A_3[5, 34] \;\; A_4[34, 8] \;\; A_5[8, 13] \; .$$

(a) Fill the table for $m(i, j)$ for $1 \leq i \leq j \leq 5$.



(b) Fill the table for $S(i, j)$ for $1 \leq i \leq 4$ and $2 \leq j \leq 5$.



(c) Describe the optimal multiplication order by adding parenthesis in the right places.

$A_1 \quad A_2 \quad A_3 \quad A_4 \quad A_5$

2. Greedy is not optimal for the multiplying n matrices problem.

   (a) Define a greedy rule to find the last multiplication for the multiplying n matrices problem.

   (b) What is the complexity of the algorithm implied by your greedy rule? Explain.

   (c) Construct an example (a sequence of matrices) for which your greedy algorithm fails to find the optimal multiplication order.

3. Let $X = (x_1, \ldots, x_n)$ be a sequence of n real positive numbers.

   The set of indices $I \subseteq \{1, \ldots, n\}$ is independent if $|i - j| > 1$ for any two indices $i, j \in I$. That is, there is no index $k \in \{1, \ldots, n-1\}$ such that both k and $k+1$ are in I.

   The weight of a set $J \subseteq \{1, \ldots, n\}$ is the sum of the corresponding x_i: $w(J) = \sum_{i \in J} x_i$.

   An independent set I is **maximum** if $w(I)$ is greater or equal to $w(J)$ for any independent set J.

   The goal is to find such a set for a given sequence X.

   (a) Find the maximum independent set of the following sequence: $(13, 34, 22, 12, 17, 40, 15, 11, 23)$.

   (b) Prove that if $x_i > x_{i-1} + x_{i+1}$ then $i \in I$.

   (c) Prove that there is no index $2 \leq k \leq n-1$ such that $k-1$, k, and $k+1$ are not in I.

Assume the following greedy solution. Initially let I be empty and $J = \{1, \ldots, n\}$. The set J is the set of indices that are not in I and differ by more than 1 from any index in I. That is, $|i - j| > 1$ for any indices $i \in I$ and $j \in J$. Then as long as J is not empty:

(i) **transfer** the index $j \in J$ with the highest value of x_j to I.

(ii) **omit** the indices $j - 1$ and $j + 1$ from J if they are in J.

(d) Show an example of a positive sequence for a **small** n for which this greedy algorithm fails to produce an optimal solution.

(e) Prove that this greedy algorithm finds a set I whose weight is at least half of the weight of a maximum independent set.

Define the following dynamic programming solution. Let $W(k)$ be the weight of the maximum independent set for the prefix sequence $X_k = (x_1, \ldots, x_k)$ for $1 \leq k \leq n$.

Note that $W(n)$ is the answer to the original problem.

(f) Define the initial values and the recursive formula for $W(k)$.

(g) Describe a bottom-up dynamic procedure to compute $W(\cdot)$.

Let $B(k)$ be **TRUE** if the index k belongs to the maximum independent set for the prefix sequence X_k.

(h) Modify your dynamic programming to compute the function $B(\cdot)$ as well. Describe only the modification.

(i) Describe an efficient procedure to find a maximum independent set for the sequence X based on the function $B(k)$.

(j) What is the complexity of the modified dynamic programming as a function of n? Justify your answer.

(k) What is the complexity of your procedure as a function of n? Justify your answer.