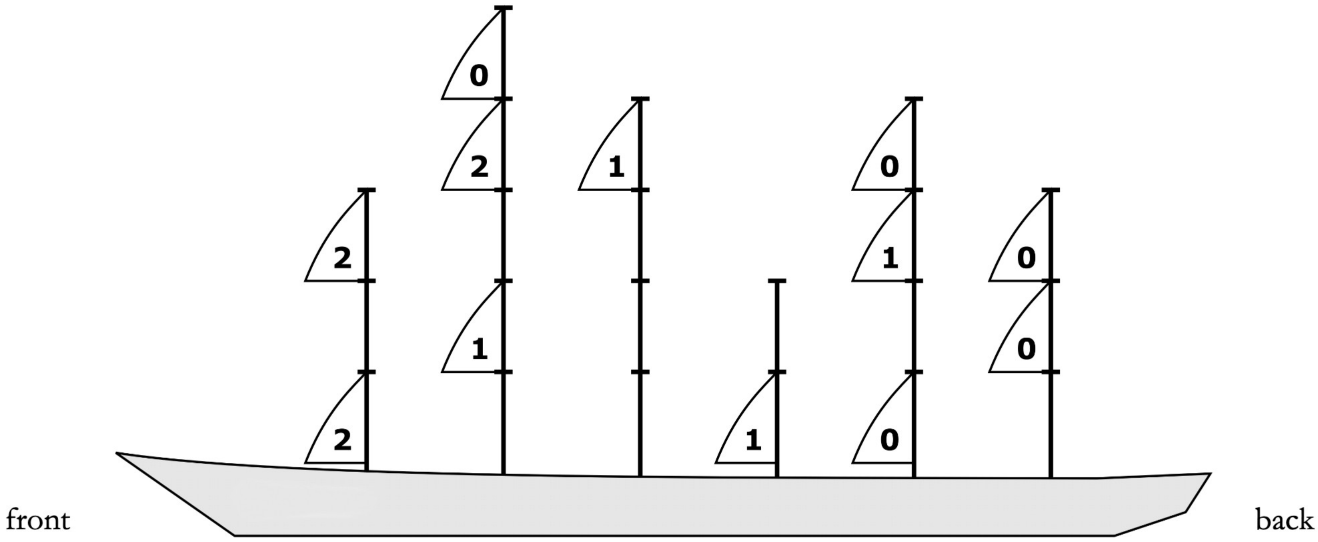
University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
S. Faour, M. Fuchs, A. Malyusz

Algorithm Theory
Sample Solution Exercise Sheet 2

Due: Friday, 10rd of November 2023, 10:00 am

Exercise 1: Migrating Pirates *(10 Points)*

The great pirates of the seven seas have decided to travel to unknown seas. For this purpose they have decided to create a new ship with N masts (poles). Every mast is divided into unit sized segments - the height of a mast is equal to the number of its segments. Each mast is fitted with a number of sails (flags) and each sail exactly fits into one segment. Given a distribution of the sails on the masts you can calculate the total inefficiency of this configuration. Which is calculated in the following way. For every flag you calculate how many flags are behind (right to) it at the same height, this gives you the inefficiency for a given sail. You add up all of these individual inefficiencies, and you get the total inefficiency. For the problem we are given the following numbers: N pairs of numbers (a_i, b_i), where a_i is the height of the ith mast and b_i ($a_i \geq b_i$) is the number of sails on the ith mast. Your task is to distribute the sails on the masts, so the total inefficiency is minimized.



This ship has 6 masts, of heights 3, 5, 4, 2, 4 and 3 from front (left side of image) to back.
This distribution of sails gives a total inefficiency of 10. The individual inefficiency of each sail is written inside the sail.

(a) Solve the problem if every mast hast the same height, in other words $\forall i, j \; a_i = a_j = C$, where C is a constant.
Hint: Look at the levels of the mast. How should the flags be distributed? *(10 Points)*

(b) *Bonus Points Problem:* Find a greedy algorithm for the original problem and prove that it is optimal. (Currently we do not know the proof.) *(10 Points)*

Sample Solution

(a) The greedy algorithm is the following. In the general case sort into an increasing order by the masts by height a_i. If you have already distributed the sails on the first $n-1$ masts. Then place the flags on the nth mast in such a way that the increase of inefficiency is minimized, basically try to distribute the flags such that at every given level the amount of sails is close to each other. Let us denote the number of flags on the ith level c_i. Now we need to prove that the greedy is optimal. We can show the following theorem.

Theorem 1. *In the optimal solution $\forall i,j$ $|c_i - c_j| \leq 1$ (For the rest of the proof let us call it property A)*

Proof. Let us indirectly assume that this is not true and as such take a counterexample where the condition does not hold. Take a pair where the condition does not hold ($|c_i - c_j| \geq 2$) in the optimal solution w.l.o.g assume $c_i > c_j$. Then increasing c_j by one and decreasing c_i by one we maintained the conditions to be a solution but the objective value strictly decreased. The new objective value is $(c_i - 1)^2 + (c_j + 1)^2 = c_i^2 + c_j^2 + 2(c_j - c_i + 1)$, where $c_j - c_i + 1$ is smaller than 0 and as such the objective value strictly decreased. This is a contradiction. □

Our greedy algorithm fulfills property A. So all we need to prove now is that this property A also implies optimality.

Theorem 2. *Property A implies optimality.*

Proof. We show that every solution that fulfills property A looks the same, that is has either x or $x+1$ flags in a row and has n_1 rows with x flags and n_2 rows with $x+1$ flags. We know the following facts about our variables:

(a) $n_1 + n_2 = C$

(b) $n_1 x + n_2(x+1) = Cx + n_2 = \sum_{i=1}^{N} b_i$

Let us assume contradiction that is $\exists y \neq x$ and m_1, m_2 numbers fulfilling the above two facts, but the objective value is not the same as for the x value. Let us assume that $y < x$ then it has to be true that $Cy + m_2 = Cx + n_2$ from this we get that $C(x-y) = m_2 - n_2$. Because every variable is a natural number $C(x-y) \geq 1$ because of this $m_2 = C$ and $n_2 = 0$ so $x = y$, but this is a contradiction. The $x < y$ case is the same. □

Exercise 2: More about the Matroid Greedy algorithm *(10 Points)*

In the following problems, we assume that you are given a matroid $M = (S, F)$, a cost function $c : S \to \mathbb{R}_+$, and an independence oracle, that is You have a black box function that gives you back whether a set $A \subset S$ is independent or not. Your algorithms should run in polynomial time in $|S|$ and the number of Oracle calls. Additionally, you can use the following definition and theorem in your proofs.

Definition 0.1. We call a maximal independent set in M a basis or a base of M.

Definition 0.2. A minimal dependent set in an arbitrary matroid $M = (S, F)$ will be called a circuit(cycle) of M. Here a minimal dependent set means if you take away any element from it, it becomes independent. (If you have a graph you would call this a minimal cycle.)

Theorem 3. *If you are given a base B of the matroid, and you add an element $e \notin B$ to B, there will be a unique cycle in $B \cup e$ which contains e. If you remove any element of this cycle from $B \cup e$ you get a base of the matroid.*

(a) Prove that if every value of $c(s), s \in S$ is unique, then you have a unique maximum weight base. *(3 Points)*

(b) Given two cost functions for the base elements of the matroid c_1, c_2. Find a base that has the maximum weight according to c_1, and among these have the maximum weight according to c_2. You also need to prove correctness. *(4 Points)*

(c) Use the previous algorithm to algorithmically solve the following problem. Given an independent set G decide if it can be extended to a maximum weight base. You also need to prove correctness. *(3 Points)*

Sample Solution

For all the proofs we will use the following greedy algorithm. Sort the elements into a descending order according to a given weight c. Maintain an independent set of B throughout the algorithm. Go through the sorted list one by one and add an element to B if it remains independent.

(a) Let B_1 be the base we got from our greedy algorithm. Assume contradiction there exists a base B_2 such that $c(B_1) > c(B_2)$. Let us look at the first case in the greedy algorithm where an element is added to B_1, but it is not in B_2, let this element be $e \in S$. Important to note that it can not happen that we added it to B_2 but not B_1 because we add an element to B_1 if it remains independent and until element e, B_1 is fully in the B_2 set. So if we add e to B_2 we create a unique cycle in B_2 according to the useable theorem.

Theorem 4. *$\exists f \in B_2$ element in the unique cycle such that $c(f) < c(e)$. So we can remove from B_2 element f and add e.*

Proof. Assume contradiction. In this case, every element in the cycle is also in B_1, but this would mean there is a cycle in B_1 which is a contradiction. □

You could have also just copied the same proof from the minimum spanning tree version `https://math.stackexchange.com/questions/352163/show-that-theres-a-unique-minimum-spanning-tree-if` which is the same as this.

(b) Sort the weights by c_1 then amongst elements that are equal according to c_1 sort by c_2. We claim this is optimal. We use the same kind of argument as in the previous problem. Assume contradiction there exists a base B_2 such that $c_1(B_1) = c_1(B_2)$ and $c_2(B_1) > c_2(B_2)$. Let e be the first element in the algorithm where a given e element is in B_1, but it is not in B_2. Add e to B_2 thus creating a unique cycle.

Theorem 5. *$\exists f \in B_2$ element in the unique cycle such that $c_1(f) = c_1(e)$ and $c_2(f) < c_2(e)$. So we can remove from B_2 element f and add e.*

Proof. We can prove it the same way as the above problem. First it is true that $\exists f \in B_2$ element in the unique cycle that it not in B_1. It also needs to be true that $c_1(f) = c_1(e)$ otherwise we could exchange f to e in B_2 achieving a base with larger weight then the original B_2. As such the only possibility is that $c_1(f) = c_1(e)$ and $c_2(f) < c_2(e)$. Just as the claim needed it. □

So we could exchange f to e and achieve a better value, thus contradiction.

(c) Use the characteristic function of G, a function that is 1 on elements of G and 0 otherwise. Let $c_1 = \chi_G$ and $c_2 = c$. If the given weight of the solution equals the max weight base then we have it is possible to expand G into a maximum weight base otherwise not.