Lab03-Greedy Strategy

CS214-Algorithm and Complexity, Xiaofeng Gao, Spring 2019.

∗ If there is any problem, please contact TA Mingran Peng.

∗ Name: Yitian Ma Student ID: 517021910467 Email: mkochab@sjtu.edu.cn

1. Suppose there is a street with length n, described by an array $A[1...n]$ where $A[i] = 1$ means that there is a house at position i and $A[i] = 0$ means position i is vacant.

   According to some law, every house must be protected by fire hydrant. If a fire hydrant is placed at position i, then all houses at postion $i-1, i, i+1$ will be considered protected. Note that hydrants can be placed at the same place with a house.

   Using what you learnt in class, please design an algorithm that computes the minimum number of hydrants needed to protect all houses. You need to write pseudo code, analyze the time complexity, and prove its correctness.

   Proof. We start from position 1 and check each position sequentially. Our strategy to place the fire hydrants is that each time we find an unprotected house at position i, we will place a fire hydrant at position $i+1$ (trying to "fully utilize" the protection space). Therefore, the position $i+2$ is also under protection. So we will continue the procedure from position $i+3$.

Algorithm 1: Greedy Placing the Fire Hydrant

Input: An array $A[1, \cdots, n]$

Output: num, the minimum number of fire hydrants

```
num ← 0, i ← 1;
while i ≤ n do
    if A[i] = 0 then
        i ← i + 1;
    else
        num ← num + 1;
        i ← i + 3;
return num
```

Time Complexity: Obviously, we will check n positions at most. So the time complexity is $O(n)$.

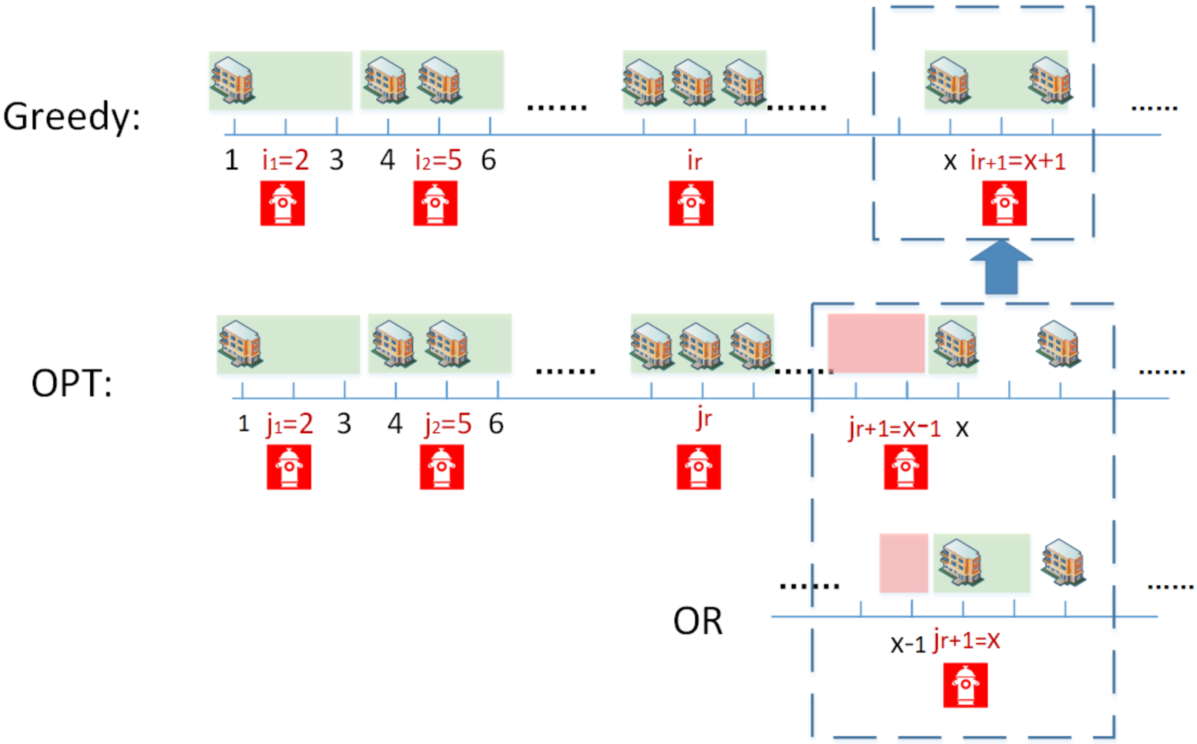
Correctness: Let's prove that greedy algorithm is correct by contradiction.

Assume that the greedy algorithm doesn't provide the optimal solution.

Let $i_1, i_2, \cdots, i_k$ $(i_1 < i_2 < \cdots < i_k)$ denote sets of positions selected by greedy.

Let $j_1, j_2, \cdots, j_m$ $(j_1 < j_2 < \cdots < j_m)$ denote sets of positions in an optimal solution with $i_1 = j_1$, $i_2 = j_2$, $\cdots$, $i_r = j_r$ for the largest possible value of r. Then we know that position 1 to $i_r + 1 (= j_r + 1)$ is now protected.

Now consider the $(r+1)^{th}$ fire hydrant.



Suppose the next(closest to i_r) house without protection is at position x. Obviously, from position 1 to $x-1$, there is no need for any other protection. Meanwhile, to put the position x house under protection, we have to place a fire hydrant at position $x-1$, x or $x+1$. According to our design, the greedy algorithm always chooses position $x+1$. As the figure illustrates, if OPT chooses position $x-1$ or x, there will be unnecessary waste of space and of course we can replace the choice with $x+1$. After the shift, OPT' is still feasible and optimal, but it contradicts the maximality of r.

□

2. (a) Given a set A containing n real numbers, and you are allowed to choose k numbers from A. The bigger the sum of the chosen numbers is, the better. What is your algorithm to choose? Prove its correctness using **Matroid**.

Remark: This is a very easy problem. Denote $\mathbf{C}$ be the collection of all subsets of A that contains no more than k elements. Try to prove $(A, \mathbf{C})$ is a matroid.

Proof. Denote $\mathbf{C}$ be the collection of all subsets of A that contains no more than k elements. If $D \subset E$, $E \in \mathbf{C}$, obviously $|D| < |E| \leq k$ and $D \subset A$. So $D \in \mathbf{C}$ is true. Thus, $(A, \mathbf{C})$ is an independent system. We apply the Greedy-MAX algorithm to $(A, \mathbf{C})$. We define the associated strictly positive function $c(\cdot)$ to each element $x \in A$. In this case, we may have $c(x) = x - min\{x|x \in A\} + 1$. The function extends to subsets of A by summation: $c(B) = \sum_{x \in B} c(x)$.

Algorithm 2: Greedy-MAX

Input: k, the number of elements to choose
Output: A set I

Sort all elements in A into ordering $c(x_1) \geq c(x_2) \geq \cdots \geq c(x_n)$
$I \leftarrow \emptyset$
for $i \leftarrow 1$ ***to*** n **do**
  if $|I \cup \{x_i\}| \leq k$ **then**
    $I \leftarrow I \cup \{x_i\}$;
return I

Correctness: Firstly we try to prove $(A, \mathbf{C})$ is a matroid.

- Hereditary: If $D \subset E$, $E \in \mathbf{C}$, obviously $|D| < |E| \leq k$ and $D \subset A$. So $D \in \mathbf{C}$ is true.
- Exchange Property: Consider D, $E \in \mathbf{C}$ with $|D| < |E|$. Since $|D| < |E| \leq k$, $|D| \leq k - 1$. Choose an element x in E that is not contained in D. $|D \cup \{x\}| \leq k$ and $D \cup \{x\}$ is a subset of A. So $D \cup \{x\} \in \mathbf{C}$ is proved.
  So $(A, \mathbf{C})$ is a matroid. Previously we used the Greedy-MAX algorithm to find a set $I \in \mathbf{C}$ maximizing the weight $c(I)$. We know that if $(A, \mathbf{C}, c)$ is a weighed matroid, then Greedy-MAX algorithm performs the best solution. So I is the best solution we want.

$\square$

(b) Consider that $B_1, B_2 ... B_n$ are n disjoint sets, and let d_i be integers with $0 \leq d_i \leq |B_i|$. Define $\mathbf{C}$ is a collection of set $X \subseteq \cup_{i=1}^{n} B_i$, where X has such property:

$$\forall i \in \{1, 2, 3 ... n\}, |X \cap B_i| \leq d_i$$

Prove that $(\cup_{i=1}^{n} B_i, \mathbf{C})$ is a matroid.
Remark: You may easily find that the matroid in (a) is a special case of matroid in (b).

Proof.

- Hereditary: If $D \subset E$, $E \in \mathbf{C}$, $\forall i \in \{1, 2, 3 ... n\}$, $|D \cap B_i| \leq |E \cap B_i| \leq d_i$. So $D \in \mathbf{C}$ is true.
- Exchange Property: Consider D, $E \in \mathbf{C}$ with $|D| < |E|$. As $|D| < |E|$, at least for one set B_x there exists $|D \cap B_x| < |E \cap B_x| \leq d_x$ (Otherwise we will have $\forall i \in \{1, 2, 3 ... n\}, |D \cap B_i| \geq |E \cap B_i| \Rightarrow |D| \geq |E|$).
  Choose an element $x \in B_x$ in E that is not contained in D. Initially we have $|D \cap B_x| < |E \cap B_x| \leq d_x$, so $|D \cap B_x| \leq d_x - 1$. For the set B_x, $|(D \cup \{x\}) \cap B_x| = |D \cap B_x| + 1 \leq d_x$. Since $B_1, B_2, \cdots, B_n$ are n disjoint sets, x doesn't belong to any other sets except B_x. For all the other sets except B_x, $|(D \cup \{x\}) \cap B_i| = |D \cap B_i| \leq d_i$. Therefore we have proved $D \cup \{x\} \in \mathbf{C}$.
  So we can conclude that $(\cup_{i=1}^{n} B_i, \mathbf{C})$ is a matroid.

$\square$

Remark: You need to include your .pdf and .tex files in your uploaded .rar or .zip file.