Lab03-Greedy Strategy

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

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

∗ Name: 田雪飞 Student ID: 515030910347 Email: 13487426939@qq.com

1. **Proof.** The following is the proof process.

• Greedy algorithm:
From $1 \leftarrow n$,if $A[i] = 1$, we can put a fire hydrant in position $i + 1$,then $i = i + 3$, skip i to $i + 3$ and continue to check;
if$A[i] = 0$,we do not put anything in here and skip to $i + 1$,and continue to check;

• pseudo code:

Algorithm 1: Greedy algorithm

Input: An array $A[1, \cdots, n]$
Output: minimum number of hydrants

```
i ← 1;
count ← 0;
while i ≤ n do
    if A[i] == 1 then
        if i == n  put a hydrant in position n;
        if i! = n  put a hydrant in position i + 1;
        count = count + 1;
        i = i + 3;
    if A[i] == 0 then
        i = i + 1;
return count;
```

• Proof:
we assume that an optimal algorithm which skip i to put a hydrant in position $j = i + 1$: we prove that we can transform this optimal strategy to greedy algorithm;
we assume that optimal and greedy algorithm have same strategy before position i;

optimal algorithm: $\cdots \underset{\text{i}}{1} \quad \underset{\text{j=i+1}}{\underline{A[i+1], A[i+2], A[i+3]}}$

$\Downarrow$ we need a hydrant to cover $A[i]$

optimal algorithm: $\cdots \underset{\text{i}}{\underline{1}} \quad \underset{\text{j=i+1}}{\underline{A[i+1], A[i+2], A[i+3]}}$

$\Downarrow$ this is same as following strategy.

optimal algorithm : $\cdots \underline{1, A[i+1], [i+2]}, \underline{A[i+3]}$;

greedy algorithm : $\cdots \underline{1, A[i+1], [i+2]}, A[i+3]$;

we can see optimal strategy is not better than greedy algorithm.when j=i+2,i+3 greedy is also not worse than optimal.
proof end. □

2. **Proof.** The following is proof process:
   (a) •we can sort the n real number to be nonincreasing sequence, and if $M \cup x \in \mathbf{C}$,we put x in M until i find the biggest sum.

   •According to the definition of uniform matriod:
   A is a set with n elements, denote **C** be the collection of all subsets of A that contains no more than k elements.
   Proof:
   Hereditary: if M $\subset$ N and N $\in$ **C**, obviously, because N is not more than k elements, so N can not more than k elements,then N $\in$ **C**.
   Exchange property: we assume that M,N $\in$ **C** and $|M| > |N|$;we know M is not more than k elements, so N is not more than k elements; meanwhile,there must exist an element in M and not in N ($x \in M$ and $x \notin N$); and we know $N \cup \{x\}$ is not more than k elements. so $N \cup \{x\} \in \mathbf{C}$.
   so, $(A, \mathbf{C})$ is a matriod.

   (b) Proof:
   Hereditary: if M$\subset$ N and N$\in$ **C**; according to the property of N, then $\forall i \in \{1, 2, 3...n\}$, $|N \cap B_i| \leq d_i$; $M \subset N$, so, $|M \cap B_i| < |N \cap B_i| \leq d_i$, so, $\forall i \in \{1, 2, 3...n\}, |M \cap B_i| \leq d_i$, then M$\subset$ **C**.
   Exchange property: we assume that M,N $\in$ **C** and $|M| > |N|$; then $\forall i \in \{1, 2, 3...n\}$, $|M \cap B_i| \leq d_i$ and $|N \cap B_i| \leq d_i$;
   (1). $|N \cap B_i = d_i$, but we know,X is subset of $\cup_{i=1}^{n} B_i$, we assume that X=$\cup_{i=1}^{n} B_i$, then $|X \cap B_i| = d_i$, so $N = B_i$; at this case, there must exist an element x which is in M but not in N; so $(N \cup \{x\}) \cup B_i = d_i$, this's still satisfying condition. so $N \cup \{x\} \in \mathbf{C}$.
   (2). if $|N \cap B_i| < d_i$,obviously, $|M| > |N|$; there must exist an element x in M and not in N; we put x into N,and $(N \cup \{x\}) \cap B_i \leq d_i$; so $N \cup \{x\} \in \mathbf{C}$.
   so, $(\cup_{i=1}^{n} B_i, \mathbf{C})$ is a matroid.

$\square$

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