算法设计 HW03

吴硕 522030910094

2024 年 4 月 21 日

Q1 The GPT is right. As it is a regret heap. We insert p_i twice to ensure that we can regret.

Considering the first time we insert p_i, and the heap top is p_j, and at last the num in heap is the price we sold out double or the day we do nothing.

The best solution must end with no stock in our hand if not we do nothing in the buying stock day, we can get higher profit.

Let p_i is i step we insert into heap, and p_j is the top of the heap. If we pop out p_j , we get profit $p_i - p_j$. However, if it is wrong, we should not sell at this day. We should sell at p_k, which means p_k is larger than p_i and all two p_i will be pop out, and we shuld buy at p_i, and sold it at p_m, so the profit we can get is actually $p_i - p_j + p_k - p_i + p_m - p_i = p_k - p_j + p_m - p_i$, which is just the best solution.

What's more, the first step, is just the same. If we buy buy or do nothing, it will always be pushed into the heap, so GPT's first step is best solution.

As a result, GPT' s every step is the best solution, so the algorithm is correct.

Q2 First, we design a function $\phi(n)$, which means after n steps how many 1 in the integer. C, which means the real cost of ADD. A, which means the amortized cost of ADD. And assume at the i step $A_i = C_i + \phi(i) + \phi(i-1)$.

As a result, after n steps, the total cost is

$$\sum_{i=1}^{n} A_i = \sum_{i=1}^{n} C_i + \phi(n) + \phi(0)$$

And $\phi(0) = 0$, so $A > C$, which means the amortized cost is larger than the real cost.

Then consider every step :

if the i step just make the last 0 to 1, so $\phi(i)+\phi(i-1) = 1$, and $C_i = 1$, so $A_i = 2$

if the i step make the last 1 to 0, assume it makes m 1 to 0 togather and one 0 to 1, so $\phi(i) - \phi(i-1) = 1 - m$, and $C_i = m + 1$, so $A_i = 2$

As a result, $A_i = 2$ is always correct, so the amortized cost is $O(1)$.

Q3 (a) If there is A and B are all maximal, but$|A| < |B|$

By exchange property, we can find $x \in B \setminus A$ and $A \cup \{x\} \in I$ so A is not maximal, which is a contradiction.

As a result $|A| = |B|$, and maximal independent sets are of the same size

(b) First, hereditary property. $\forall F \in S$, F does not contain any circle. As a result, any $F1 \subset F$, $F1$ does not contain any circle, so $F1 \in S$. So, hereditary property is satisfied.

Second, exchange property. Assume $A, B \in S$, and $|A| < |B|$. We can divide A into some different connected components $A1, A2, ..., An$. For every component Ai, we can find a same component in B with the same points. As a result, we can divides all edges in B into two parts: one part is in components, the other is the edges between components.

Considering the same part, because B does not contain any circle, so the number of edges in the same part is less than $|Ai|$. As a result, all edges in components are less than $|A|$, so we can find a edge x in B which is not in A, and it conects two different components. So, we can add it into A and connect two components and do not contain a circle. So $A \bigcup \{x\} \in S$, so exchange property is satisfied.

As a result, S is a matroid.

(c) Assume there is a maximal independentset A with the maximal weight and does not contain x. By the algorithm, $\{x\}$ must in M.

So, by the exchange property, we can continueously exchange the elements in A with$\{x\}$, and the number of $|\{x\}|$ will continueously increase.

Finally, $|\{x\}| = |A|$. However, considering the weight, the weight of $\{x\}$ is larger than A, since x is the maximal and the other elements in $\{x\}$ are all in A. So, the weight of $\{x\}$ is larger than A, which is a contradiction. So A does not exist. So the maximal independent set with maximal weight must contain x.

(d) First, we can prove that at any time, S is always $\subset$ some maximal independent set with maximal weight.

Assume it is wrong. First, assume after one step, S' is no longer $\subset$ some maximal independent set with maximal weight. $S' = S \cup \{y\}$, and S is $\subset$ some maximal independent set T with maximal weight.

And $|S'| < |T|$. By the exchange property, we can find a element x in T but not in S'. So we can add x into S', continue until $|S'| = |T|$. Let $A = S' \cap T$, so $S' \backslash A = \{y\}$. We set $T \backslash A = \{x\}$

So, $w(y) >= w(x)$ by the algorithm we choose the biggest weight element into S, so $w(S') >= w(T)$, which is a contradiction.

So , at any time, S is always $\subset$ some maximal independent set with maximal weight. So finally, S is the maximal independent set with maximal weight.

(e) Let $M = (U, S)$ where $S = \{T \subset U|$ all vectors in T are linearly independent$\}$

Firet, we can prove that M is a matroid. hereditary property, if $T \in S$, and $T1 \subset T$, all vectors in T are linearly independen, all vectors in $T1$ are linearly independent, so $T1 \in S$.

exchange property, if $A, B \in S$, and $|A| < |B|$, $dim(A) < dim(B)$, so there must be a vector $x \in B \setminus A$, and $A \cup \{x\}$ is still linearly independent. So $A \cup \{x\} \in S$.

As a result, M is a matroid.

So, we can use the algorithm in (c) to solve this problem, and the correctness is proved in (d).

Next we prove the time complexity. Sort is $O(nlogn)$, and the for loop is $O(n)$, to check the linearly independent, we can use the Gaussian elimination, which is $O(n^3)$. So the total time complexity is $O(n^4)$.

Q4 Difficluty: 3 (except the first problem)

The first problem is 5, as I do not konw how to prove the correctness of the algorithm exactly. Collaborator:Zhihu and 蒋松霖.