1 Rope Renting Problem

1.1 Introduction

Here is a description of the rope renting problem. Every morning, some customers come to rent ropes for some days. The requested length of a rope is always a power of 2. To fullfill a request of length k, we need to give the customer a rope whose length is at least k.

If a requester rents a rope for d days, she will return the rope by the end of the d-th day. This returned rope maybe rented out the following morning to other customers.

Initially there is a "main rope" of infinite length. We define the cost of an algorithm to be the total length of the ropes that are cut off from the main rope. The goal of the online algorithm is to minimize the cost.

1.2 A_{GRD}: A greedy online algorithm

A_{GRD} is a greedy algorithm that always gives a customer the requested length. When a customer wants a rope of length $k = 2^d$, A_{GRD} uses the following strategy:

- If A_{GRD} has a returned rope of length exactly k, it gives this rope to the customer.
- Otherwise, if there is at least a returned rope of length larger than k, A_{GRD} ***splits*** the shortest such rope into two equal parts repeatedly, until it has a length k rope. It gives the length k rope to the customer.
- Otherwise, A_{GRD} ***cuts*** a length k rope from the "main rope" to give to the customer.

We use A_{GRD} to represent this greedy online algorithm. We use C_{GRD} to represent its cost.

1.3 A_{OPT}: A restricted optimal algorithm

We compare A_{GRD} with an off-line algorithm that we call A_{OPT}. The restrictions to A_{OPT} are that the length of any piece that A_{OPT} cuts from the main rope should be a power of 2, and the length of any rope that A_{OPT} hands out to the customer should also be a power of 2. The cost of A_{OPT} is C_{OPT}.

When a customer requests a rope of length $k = 2^d$, A_{OPT} may do zero or more of the following operations, before the algorithm hands out a rope of length $\geq k$ to the customer.

- ***cut***: cut from the "main rope" a section of length k, which is a power of 2.
- ***split***: split a rope (that has been cut from the main rope) into two equal pieces.

Without loss of generality, we assume that if A_{OPT} splits a rope in step i, it will hand out one smallest section to the reqeuster of step i. (Otherwise, we can delay the optimal algorithm's splitting.) We aslo assume that if the A_{OPT} cuts a piece of rope from the main rope, it always immediately hand it out to the requester.

Note that A_{OPT} may hand a customer a rope longer than the requested length.

1.4 Properties of A_{GRD} vs A_{OPT}

Theorem 1 $C_{GRD} < 3 * C_{OPT}$.

For each algorithm, let's initialize the units of the main rope with consecutive integers, starting from 1. The name of each unit remains unchanged even as pieces are cut off from the main rope and as those pieces are further subdivided.

We give unique labels to each piece of the rope cuf off by $A + OPT$. If the current head of the main rope is unit i, and the algorithm cuts length k from the main rope, we label the section that is cut off from the main rope with tuple (j, k). If a rope r with label (i, k) is split in half. We label the resulting two pieces $(i, k/2)$ and $(i + k/2, k/2)$. For each rope r that is not the main rope, we let $\boldsymbol{l(r)}$ denote the length of the rope. If $r = (j, k)$, then $l(r) = k$.

As an example, if rope r_1 is the first piece cut from the main rope with length 8, then r_1's label is $(1, 8)$. Similarly, if second rope r_2's length is 16, then the label of r_2 is $(9, 16)$. In the remaining of the paper, for simplicity, we interchange the use of a rope and its label.

Let's assume that both A_{OPT} and A_{GRD} proceed in rounds. Each round consists of a customer request, followed by some operations taken by the algoirthm, followed by handing out a rope to the requester.

Let's lock the steps of both agorithms, and examine the ropes cut off by both algorithms at the end of each step i.

Let $\boldsymbol{P_i}$ be the set of finite length ropes that A_{OPT} has at the end of step i. Some of the ropes in P_i may be in customers hands, and some maybe in A_{OPT}'s hands.

Let $\boldsymbol{U_i} = \{1, 2, ..., L_i\}$ denote the set of the units cut off by A_{GRD} at the end of step i.

We will construct a funtion $\boldsymbol{f_i} : U_i \to P_i$ such that f_i maps each unit of the rope that is cut off by A_{GRD} to a piece of A_{OPT} rope at the end of each step i.

For any $r \in P_i$, define

$$g_i(r) = |\{j : j \in U_i, f_i(j) = r\}| \ .$$

Then $g_i(r)$ is the number of A_{GRD}'s units that is mapped to rope r of A_{OPT}. We have $C_{GRD} = \sum_{r \in P_i} g_i(r)$. We will show that after each step i, we can construct f_i, such that, for any $r \in P_i$, $g_i(r) < 3l(r)$.

Lemma 2 *There exists a mapping* $f_i : U_i \to P_i$*, such that* $\forall r \in P_i$*,*

$$g_i(r) < 3l(r) \ .$$

Proof.

For each $j \in U_i$, let's use $\text{rope}_i(j)$ to denote the rope that contains unit j in A_{GRD} at the end of step i. For any A_{OPT}'s rope r and any integer k, define

$$h_i(r, k) = |\{j : j \in U_i, l(\text{rope}_i(j)) = 2^k, f_i(j) = r\}| \ .$$

$h_i(r, k)$ is the total number of units, among A_{GRD}'s ropes of length 2^k that is mapped to A_{OPT}'s rope r.

First, consider step 1. We assume that the customer requested length is 2_d. Both algorithms will cut a piece of rope from the main rope and hand out the piece to the customer. Let r_a be the rope A_{OPT} cuts, and r_g be the rope that A_{GRD} cuts. We have $l(r_g) = 2_d$, and $l(r_a) \geq 2^d$. For $j = 1, \ldots, 2_d$, we set $f_1(j) = r_a$.

Then, for $i = 1$, $g_i(r_a) = 2_d \leq l(r_a) < 3l(r_a)$. We also note that f_i satisfies the following 3 invariants, for any rope $r \in P_i$:

- 1. If at step i, A_{OPT}'s rope r is rented out to customar a, and A_{GRD}'s rope t is rented out to the same customer, then, each unit of t is mapped to r under f_i. That is, for any unit j such that $\text{rope}_i(j) = t$, we have $f_i(j) = r$.

- 2. For all k that is a power of 2, such that $k < l(r)$, we have $h_i(r, k) <= 2k$. This invariant says, consider all A_{GRD} ropes of length exactly k, the number of units of these ropes that is mapped to r is at most $2k$.

- 3. $g_i(r) < 3l(r)$.

Note that the three invariants are meaningful even if f is only partially defined on U_i, i.e, $\exists j \in U_i$ such that $f_i(j)$ is not defined.

We will assume that at the end of step $i-1$, there is a fully defined mapping $f_{i-1} : U_{i-1} \to P_{i-1}$ which satisfies the three invariant. For step i, we will construct $f_i : U_i \to P_i$ based on f_{i-1} and based on the operations taken by both A_{OPT} and A_{GRD} during step i.

We first, set f_i partially according to f_{i-1}, i.e. for all $j \in U_{i-1}$, let $f_i(j) = f_{i-1}(j)$. We see that if $U_i \neq U_{i+1}$, then f_i is not fully defined. However, the three invariants hold for this partially defined f_i.

Next we will modify f_i based on each operation performed by A_{OPT} in step i. Then we modify f_i based on each operation performed by A_{GRD} in step i. We show that the three invariants still hold after each modification of f_i. (f_i may not be fully defined until we've taken all operations of A_{GRD} into account.)

Let's assument that in step i, the request length is 2_d. We consider the actions of A_{OPT} first. A_{OPT} can do several cut or split operations, before handing out a rope.

- cut. A_{OPT} may cut from the main rope a rope r of length 2^k. We do not update f_i. The the invariants have not changed.

- split. A_{OPT} may split a rope of length $r = (s, 2^{k+1})$ into two ropes $r_1 = (s, 2^k)$ and $r_2 = (s + 2^k, 2^k)$.

  Let $W = \{j : j \in U_{i-1}, f_i(j) = r\}$. We remap elements in W into either r_1 or r_2 using the following alogirthm .

  We first sort the elements in W based on the length of the ropes to which the elements belong. After sorting, let $j_1, j_2, \ldots, j_{|W|}$ be the elements of W in sorted order. If $j_x \in W$ and $j_{x+1} \in W$, then $l(\text{rope}_{i-1}(j_a)) \leq l(\text{rope}_{i-1}(j_b))$.

  For each $j = j_1, j_2, \ldots, j_{|W|}$, if after we let $f_i(j) = r_1$, all invariants still hold true, then we set $f_i(j) = r_1$; otherwise, we set $f_i(j) = r_2$.

  It is easy to see that invariant 1 holds after the remapping, since for any $j \in W$, $rope_i(j)$ cannot be handed out to any customer at this point.

  Invariant 2 holds for r_1, because for $n < l(r_1) = 2^k$, $h_i(r_1, n) <= 2n$ according to the construction of f_i.

Now consider r_2. If for $n < l(r_1)$, $h_i(r_2, n) > 2n$, then we must have $h_i(r_1, n) = 2n$. Further, we have $h_{i-1}(r, n) = h_i(r_2, n) + h_i(r_1, n) > 2n + 2n > 2n$. This is a contradiction to the fact that all invariants holds after step $i - 1$. So invariant 2 must hold for r_2.

Since $\sum_{n=1}^{k-1} h_i(r_1, 2^n) < 22^k$, and $l(r_1) = l(r_2) = 2^k$, after we have remapped $f_i(j)$ for all j such that $l(\text{rope}_{i-1}(j)) < k$, invariant 2 holds without violating invariant 3. And no matter how we assign f_i for the remaining units $j \in W$, invariant 2 does not get changed for either r_1 or r_2. Thus, for any unit j such that $l(\text{rope}_{i-1}(j)) >= 2^k$, we set $f_i(j) = r_2$ only under the condition that $g_i(r_1) = 3l(r_1)$, in which case, $g_i(r_2) = g_{i-1}(r) - g_i(r_2) < 3l(r) - 3l(r_1) = 3l(r_2)$. Thus invariant 3 holds for both r_1 and r_2.

- handout a rope. Nothing changes of f_i.

So, after modifying f_i according to A_{OPT} the invariants hold.
Now let's update f_i according to A_{GRD}'s actions:

- cut and handout. A_{GRD} cuts a piece $r = (s, 2^d)$ from the main rope and hands this rope to the customer. Let rope q be the rope that A_{OPT} handed out to customer in step i. Then for all $j = s, \ldots, s + 2^d - 1$, assign $f_i(j) = q$.

  Since A_{GRD} cuts the rope from main, it must be that, for A_{GRD}, all ropes of length 2^d or longer are currently handed out to customers. So all the units that are mapped to q are of length smaller than 2^d. Since invariant 2 holds before the new assignment of f_i, $\forall k$ such that, $2^k < l(q)$, we have $h_i(q, 2^k) <= 2(2^k)$, and $h_i(q, 2^d) = 0$. After the new assignment of f_i, we have $h_i(q, 2^k)$ unchanged for k such that $2^k < l(q)$, except $h_i(q, 2^d) = 2^d$. Thus invariant 2 holds for q. All other ropes in P_i are not affected.

  Invariant 3 also holds for q becuase $g_i(q) = \sum_{k=1}^{d} h_i(q, 2^k) < \sum_{k=1}^{d-1} 2(2^k) + 2^d < 3(2^d) = 3l(q)$.

- split until done and handout. A_{GRD} splits rope $r = (t, 2^k)$, where $k > d$ into $(k - d + 1)$ pieces: for $n = d+1, \ldots, k-1$, there is one rope of length 2^n; and there are 2 ropes of length 2^d.

  Let $r_1 = (s, 2^d)$ be the piece of rope of length 2^d that A_{GRD} hands out to the requester. Let $q \in P_i$ be the rope that A_{OPT} handed out to the reqeuster for step i.

  We first modify f_i by performing a series of "exchanges":

  For any $j \in \{s, s+1, \ldots, s + 2^d - 1\}$, if $f_i(j)! = q$, and if $\exists j', f_i(j') = q$, such that $\text{rope}_i(j')$ is the shortest rope that $l(\text{rope}_i(j')) \geq 2^d$, then, we set $f_i(j') = f_i(j)$ and $f_i(j) = q$. That is, we exchange $f_i(j)$ and $f_i(j')$'s values.

  Finally, after the exchanges, if there is still some j such that $\text{rope}_i(j) = r_1$ and $f_i(j)! = q$, set $f_i(j) = q$.

  We need to verify that for any rope $t \in P_i$, the invariants hold for step i. Note that at this point, f_i is fully defined on U_i.

  We consider invariant 1 first. In the beginning of step i, if rope r_g of A_{GRD} is in a customer's hand, then for any unit j such that $rope(j) = r_g$, $f_i(j)$ would not be modified during step i, since all units of r_g is mapped to some A_{GRD}'s rope r_a, where r_a and r_g serve the same customer.

During step i, A_{OPT} hands out rope q to the customer, while A_{GRD} hands out rope r_1 to the same customer. Since $\forall j$ where $rope_i(j) = r_1$, we have set $f_i(j) = q$, invariant 1 still holds after step i.

Now let's consider invariant 2 and 3. For any $t \in P_i$, there are 2 cases.

- Case 1: $t! = q$. Invariant 3 holds because no new units are mapped to t. Invariant 2 holds, because, before splitting, for any $j \in \{d, \ldots, k-1\}$, there cannot be two units belonging to two different ropes of length 2^j mapped to t. Otherwise, at least one of the rope of length 2^j is not handed out at the beginning of step i. As a result, A_{GRD} would not handing out a rope of length 2^j instead of rope r where $l(r) = 2^k > 2^j$. So, before splitting $h_i(t, j) <= 2^j$. Because of splitting of r, at most one rope of length 2^j is created which may have units mapped to t. Thus after splitting $h_i(t, j) <= 2(2^j)$. So invariant 2 holds.
- Case 2: $t = q$, where q is the rope A_{OPT} handed out to the customer in step i.

  Recall that for $n = d, \ldots, k-1$, there is no rope of length 2^n which has even one unit mapped to q – otherwise, A_{GRD} would not be splitting rope r which is of length 2^k. At the end of step i, there are at most two ropes of length 2^n that has any units mapped to q. Thus $h_i(r, 2^n) <= 2(2^n)$, for $n = d, \ldots, k-1$. As a result invariant 2 holds.

  For invariant 3, if after "exchange", all units of r_1 are mapped to q , then the totall units of A_{GRD}'s ropes that is mapped to q has not changed, which implies that $g_i(q) = g_{i-1}(q) < 3l(q)$. If there are some units j, $rope_i(j) = r_1$ and $f_i(j)! = q$ right after the "exchange", then at the end of step i, no unit that belongs to a rope of length at least 2^d, except rope r_1 itself, is mapped to q. Thus the total units mapped to q is at most

$$\begin{aligned}
\sum_{n=0}^{d-1} h_i(r, 2^n) + 2^d - 1 &\leq \sum_{n=0}^{d-1} 2(2^n) + 2^d - 1 &(1)\\
&< 2(2^d) + 2^d - 1 &(2)\\
&< 3(2^d) &(3)\\
&= 3l(q)\ . &(4)
\end{aligned}$$

  As a result, invariant 3 holds at the end of step i.

□

Theorem 1 is a corrallary of Lemma 2.