COMP 5711 - Advanced Algorithms Written Assignment # 4

ZHAO, Xi
12256508
xzhaoca@connect.ust.hk

December 3, 2023

SKI

(a)

We adopt the break-even algorithm [1] to solve this problem: I rent for $n-1$ days and buy skis on the morning of day n if I am still up for skiing. If I have to stop skiing during the first $n-1$ days, it costs the same as what I would pay if one had known the number of days I would go skiing. If I have to stop skiing after day n, my cost is $(2n-1)$ dollars, which is at most as twice as what I would pay if I had known the number of days I would go skiing in advance. So, it achieves a competitive ratio of at most 2.

(b)

Consider an adversary that knows the deterministic strategy and chooses the number of days to ski accordingly. If the strategy always rents for k days before buying, the adversary can choose to ski for $k+1$ days and then stop, forcing the strategy to rent for k days and then buy a ski, which costs $k+n$ dollars. When $k < n-1$, $\mathsf{OPT}= k+1$,

$$\frac{k+n}{k+1} = 1 + \frac{n-1}{k+1} > 1 + \frac{n-1}{n} = 2 - 1/n.$$

When $k \geq n-1$, $\mathsf{OPT}= n$,

$$\frac{k+n}{n} = 1 + \frac{k}{n} \geq 2 - 1/n.$$

So, no deterministic strategy can achieve a competitive ratio better than $2 - 1/n$.

(c)

Since n is an even number, let $n = 2k$ and $q = 1 - 1/k (q < 1)$. We rent the ski until day $n/2 = k$. Then with the probability $p_i = \alpha q^{2k-i}$ we buy the ski on the morning of day i if I am still up for skiing; otherwise we continue to rent until day i. Here $\sum_{i=k+1}^{2k} p_i = 1$ and thus

$$\alpha = \frac{1}{\sum_{i=k+1}^{2k} q^{2k-i}} = \frac{1-q}{1-q^k}.$$

[1] `https://en.wikipedia.org/wiki/Ski_rental_problem`

When we buy the ski in the day i, the competitive ratio is at most $\frac{n+i-1}{n} < 1 + i/n$ based on the results in (a) and (b). Therefore, the competitive ratio in expectation is at most

$$
\begin{aligned}
E <& \sum_{i=k+1}^{2k} (1 + i/n)p_i \\
=& 3/2 + \frac{1-q}{2k(1-q^k)} \cdot \frac{kq}{1-q} \\
=& 3/2 + \frac{q}{2(1-q^k)} \\
<& 3/2 + q/2 = 2 - 1/n.
\end{aligned}
$$

So, we obtain a better competitive ratio.

MG

(a)

After counting N items, there are at most $N - M$ decrements. So, at most $(N-M)/(k+1)$ decrements on each unique key, indicating the error is at most $(N-M)/(k+1)$.

(b)

Denote f_i' as the counter value of the i-th most frequent item . We have $f_i' \geq f_i - (N - M)/(k+1), 1 \geq i \geq t$. So,

$$
M \geq \sum_{i=1}^{t} f_i' \geq \sum_{i=1}^{t} (f_i - \frac{N-M}{k+1}) = \sum_{i=1}^{t} f_i - \frac{t(N-M)}{k+1} \tag{1}
$$

According to Eq. 1,

$$
N - M \leq N - \sum_{i=1}^{t} f_i + \frac{t(N-M)}{k+1} = N^{res(t)} + \frac{t(N-M)}{k+1}. \tag{2}
$$

So, $(N-M)/(k+1) \leq N^{res(t)}/(k+1-t)$. Proven!

Parallel

Since we can obtain the location of the first 1 by comparing each element once in A, where each comparison takes $O(1)$ times. Therefore, like the prefix sum problem, this problem can be simulated by a circuit of depth $D = O(\log n)$ and work $O(n)$. When there are $p = O(n)$ processors, according to the Brent's theorem, the running time can be $O(W/p + D) = O(\log n)$.