Lecture 22

Farthest In Future : Caching

October 6, 2016

1 John and the Dating Problem

John is a smart mathematician having l girlfriends. Each of his girlfriends' names start with a different alphabet. So, we can use the initial alphabet of their names instead of the complete names. **L** is the list holding the names of all the girlfriends (in form of the initial alphabets). John daily dates exactly one of his girlfriends according to a fixed sequence. The sequence of this dating is provided in the list S.

For the sake of clarification, let us take an example. Assume l = 5 and the names of John's girlfriends be Taylor, Rihana, Miley, Selena and Alan. Then, $L = \{T, R, M, S, A\}$. Now, there is a sequence S for dating. Say $S = \{R, A, M, R, S, A, R, R, R, R, T, R, M, S, A\}$

Table 1: List S of dating sequence

									0	1					
DAY	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
GIRL	R	Α	М	R	S	Α	R	R	R	R	Т	R	M	S	А

Table	2:	List	C	at	the	second	day
-------	----	-----------------------	---	---------------------	-----	--------	-----

GIRL R A

 Table 3: List L of girlfriends

 GIRL T R M S A

All the girlfriends of John are from poor background and they do not have any expensive gifts, other than the ones which John gifts them. All these ladies are interested in wearing a diamond ring and they come for dating with him only if they have a ring¹. So, if any of his girlfriends possesses a ring offered by him, only then she will come for a date, else not. Please note that every

¹Ring now on means diamond ring only

time John asks one of his girlfriend for date, she does not ask for a ring. If she already has a ring, she comes. If she does not has one, she asks.

John, though rich, but still, can afford only k < l rings. He carefully keeps a track of his girlfriends who are possessing a ring given by him, in a list **C**. So, *C* represents the set of the current holders of the *k* rings. Since, k < l, at a given day, all of his girlfriends can not be having a ring.

Now, the game starts and John starts gifting rings to his girlfriends. Till he dates with k distinct girlfriends, there is no problem, he has enough number of rings to gift them one. But as soon as the $k + 1^{th}$ girlfriend comes, he falls in problem, since he has no ring left. As an example, say k = 3, the process goes like-

- 1. R comes, John gives her a ring and dates her. $C = \{R\}$
- 2. A comes, John gives her a ring as well and dating goes well. $C = \{R, A\}$
- 3. *M* comes, John gives her a ring as well and dating goes well. $C = \{R, A, M\}$
- 4. R comes, she already has a ring. She happily joins John. C remains same.
- 5. S comes. John had only 3 rings, already given to his girlfriends. What does John do now?

In such a case, John breakups with one of his existing ring holding girlfriends, give it to S and dates with her. So, when John has no ring to offer to a new lady with whom he wants to hangout with, what he does is to break up with one of the k girls he dated with previously and take his ring back². Still, taking back ring from a girlfriend causes her a heartburn. John being kind, wants to minimize the number of heartburns but at same time he is not willing to compromise with his fixed choice of girl for dating on any day.

1.1 John Using Farthest in Future Algorithm

To minimize the number of heartburns, he uses **Farthest In Future** strategy which he recently learnt in his **Probability and Computing** class. Being a smart mathematician, John is able to practically use his knowledge of Probability and Computing in real world.

John sees the list C in which he had stored the names of k girls who are currently possessing a ring given by him. If the next girl with whom John has to

 $^{^{2}}$ Please note that since each of his girlfriends is greedy, after break up, she will happily make-up with John if he apologizes and offers her the ring back

date with is one present in the list/cache C, there is no problem. She has a ring and she will happily join him (Cache Hit). But what if this new girl is not in C? This means she will first ask John for a ring before coming. John now has to take the ring back from one of the girls in C (Cache Miss). From whom should he take the ring back such that he could reduce the number of future heartburns?

Working with the same example we took before, $L = \mathbf{T}$, \mathbf{R} , \mathbf{M} , \mathbf{S} , \mathbf{A} , l = 5. John has only two diamond rings (**k=2**). The dating order for the girls $S = \{R, A, M, R, S, A, R, R, R, R, R, M, S, A\}$

1.2 Strategy Used by John

At any stage where John wants to take back the diamond ring from one of the girl in C, he scans his dating sequence. If there is a girl in C which never appears in the sequence now on, John takes back the ring from other, since he is never going to date her again. Otherwise, he looks the sequence and picks a girl who is farthest in future in the sequence. This way, John postpones the heartburn as much as he can. In the example discussed, John takes the ring from M, since she is farthest in the future.

2 Observations to be Made in Farthest in Future Caching Policy

John being a smart student made several observations while solving his dating problem.

1. Given John suffers from the unavailability of a ring on the day p, and uses farthest in future caching policy to take ring back from a girl in C. We know that he has caused a heartburn at time p, but he will not make any heartburn till time p + k - 1 now.

T_{ab}	ില	$\Lambda \cdot$	S

DAY	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
GIRL	R	Α	Μ	R	S	А	R	R	R	R	Т	R	М	S	А

Reason: The girl in C from whom the ring is taken is the one who in the future is seen after all the remaining girls in C. Hence all k - 1 girls will be seen at least once before any other element apart from these k - 1 elements is seen. Hence there will always be a cache hit for at least next k - 1 days. There won't be any cache eviction for at least next k - 1 days.

2. John defines a new term "Phase". The sequence S is divides into phases. The first phase runs from the first girl, till the time $(k + 1)^{th}$ distinct girl is seen. As soon as this $k + 1^{th}$ distinct girl is seen, just before her in the sequence, John puts a vertical bar and ends phase 1. Similarly phase 2 starts with the encounter of this $k + 1^{th}$ distinct element and again runs till the next $k + 1^{th}$ distinct element is seen. Total number of phases are denoted by ϕ .

Revisiting our previous example Here Phases are : 1-2, 3-4, 5-6, 7-11, 12-13, 14-15, P=6 (see Table 4)

3. No matter what-ever sequence of girls John decide and he use the most optimum algorithm ie.. **FIF**, he will still make at least $\phi - 1$ heart burns. Formally stated no matter what given list S is and how optimum our algorithm is we will at least see $\phi - 1$.

It is very easy to see this. Please note that a new phase starts with a heartburn. Since, k distinct elements are in C and it is the $k+1^{th}$ distinct element which is arriving, definitely John will have to take back a ring from one of the girls in C.

 4. Competitive Ratio = Number of heartburns caused by any other algorithm Number of heartburns caused by FIF
 Question: Give a bound on this value.
 John being a lazy student leaves this proof as a homework for you.