Problem 1. Specifically, your goal is to describe an instance – a set of (s_i, t_i) pairs in which all source vertices s_i are distinct, and all destination vertices t_i are distinct – such that the maximum travel time is at least $C\sqrt{n}$. Here, the dimension of the hypercube is d, and the number of vertices is $n = 2d$.

Suppose $d = 2k + 1$ and the number of pairs are $C\sqrt{n} \approx 2^k$.

$\text{delay}(s_i, t_i) \leq \#\{R_i \& R_j \text{ interfere}\}$

For the edge between the adjacent vertices a, b, there are at most 2^k source vertices and 2^k destination vertices so that $\text{delay}(s_i, t_i) \leq 2^k - 1$. Then, for all edges,

$max_i \text{ delay}(s_i, t_i) \leq \sum_1^d 2^k - 1 \approx C\sqrt{n}$ for some constant C.

For the lower bound, since all the selected pairs are routed according to the bit fixing strategy contain the edge $a \to b$, so at least 2^k interference will occur. Thus, the max delay is $\Omega(\sqrt{n})$

Problem 2. Consider the Ski Rental Problem with a ski rental cost of \$100 and ski buying cost of \$200. Design and analyze deterministic and randomized online algorithms for this problem. What are the competitive ratios of your algorithms?

Deterministic:

Let t denotes the number of days a person goes skiing and suppose an online algorithm is that buying skis happens on day k and renting for the previous $k - 1$ days, then

$$ALG = \begin{cases} 100t, & \text{if } t < k \\ 200 + 100k, & \text{if } t \geq k \end{cases}$$

if $t < k$, $\dfrac{ALG}{OPT} = \dfrac{100t}{100t} = 1$.

if $t \geq k$. Then, optimal algorithm is buying from day 1 so that $OPT = 200$. The online algorithm should cost $100 * 1 + 200$ for renting for the first $200/100 - 1$ day(s) and then buy the ski. Thus, the competitive ratio for this deterministic online algorithm is $300/200 = 1.5$.

Randomized:

Let t be a random number of going skiing and p be the probability of buying ski at each day. Then for day 1, the cost is $200p + 100(1 - p)$ with comparative ratio being $\dfrac{200p + 100(1-p)}{100} = 1 + p$

On day 2, the cost is $200p + 200(1 - p)$ with comparative ratio being $\dfrac{200p + 200(1-p)}{200} = 1$

On day 3, the cost is $200p + 300(1 - p)$ with comparative ratio being $\dfrac{200p + 300(1-p)}{200} = \dfrac{3-p}{2}$

When the accumulated renting cost exceeds buying cost but we keep renting, the comparative ratio would increase since the optimal algorithm would be always buying the ski. Then, a better randomized algorithm in this case would be getting the buying probability when $1 + p = \dfrac{3-p}{2}$. Solving the equation we get $p = 1/3$. Thus, a randomized algorithm could be each day, we will have the probability of $\dfrac{1}{3}$ to buy the ski, so that the comparative ratio is $1 + 1/3 = \dfrac{3 - 1/3}{2} = 4/3$.

Problem 3. On day t, your algorithm receives set S_t with the list of shows available on that day. The algorithm may choose a show from the list S_t and buy a ticket to this show. The goal of the algorithm is to maximize the number of distinct shows Alice attends. Design a 2 competitive deterministic online algorithm and analyze its performance.

Let U be the universe of m distinct elements, and $S_t \subseteq U$ for $t = 1, 2, 3, ..., n$. Let x_i for $i \in [1, m]$ denotes any distinct element in the universe and $y_i = 1$ denotes that x_i being selected and 0 otherwise, so that we want $max \sum_{t=1}^{n} y_i$.

One deterministic online algorithm is simply that for each day, select any show that has not been seen before.

for all $x_i \in S_t$,
    if any $y_i = 0$,
        set one $y_i = 1$
        break
    else if all $y_i = 1$
        pass

Let k be the maximal distinct elements that the optimal offline algorithm can choose such that $k \leq min\{m, t\}$ and suppose that there are only t' days with $t' \leq k$ where the online algorithm successfully chooses a new element. Then, since it is a maximization problem, the competitive ratio is $sup\{\frac{OPT}{ALG}\} = \frac{k}{t'} = \frac{t' + k - t'}{t} = \frac{1 + \frac{k - t'}{t'}}{1}$.

Since the definition of k ensures that there exists at least one sequence of distinct elements of length k with each element chosen from each set, and the greedy approach of our online algorithm ensures that we must choose one element if there is any unchosen available element, then, the worst case would be gaining one element at time i while losing one at certain future time i'. Thus, the total miss will be less or equal to the total gain leading to $k - t' \leq t'$.

As a result, the competitive ratio $sup\{\frac{OPT}{ALG}\} \leq 2$.