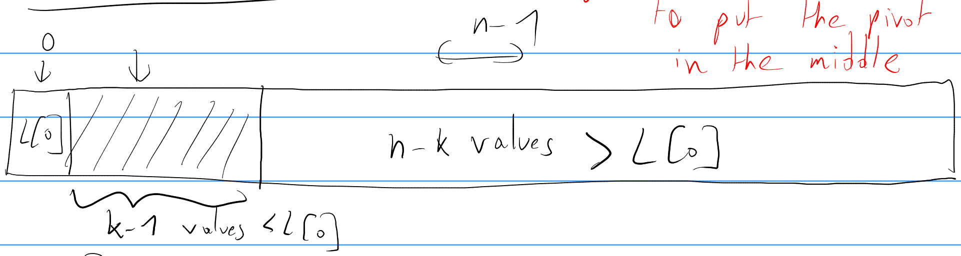
Exercise 3

We ignore the final swap to put the pivot in the middle

end:



Assume: $L[0]=k$, for some k

⚠ L is a permutation of $\{1,..,n\}$

Hint: think why only $L[i]$, $i \leqslant k$ matters for counting swaps.

Claim: $L[i]$ (for $i \leq k$) will be involved in a swap if and only if $L[i] > k$.

The number of swaps (in total) is the number of $L[i] > k$ with $i = 1,..,k$

- $L[1]$: what is the proba that $L[1] > k$?

$L[1] \in \{1,..,k-1,\underbrace{k+1,..,n}_{n-k \text{ values } > k.}\} \rightarrow$ set of size $\boxed{n-1}$

Proba $= \frac{n-k}{n-1}$

- $L[2]$: what is the proba that $L[2] > k$?

$L[2] \in \underbrace{\{1,..,n\} \setminus \{L[0], L[1]\}}_{S_2}$ $n-2$

S_2 : how many values $> k$? $\leq n-k$

The possible values $> k$ are $\{k+1,..,n\} \setminus \{L[1]\}$ of size $\leq n-k$

Proba $\leq \frac{n-k}{n-2}$

- $L[i]$: $L[i] \in \{1,..,n\} \setminus \{\overbrace{L[0],..,L[i-1]}^{i \text{ values}}\}$ $\rightarrow$ set of size $n-i$

Proba $\leq \frac{n-k}{n-i}$

$\mathbb{E}\left[\#\text{ of swaps}\right](\text{on average}) = \sum_{i=1}^{k-1} P\left[L[i] > k\right]$

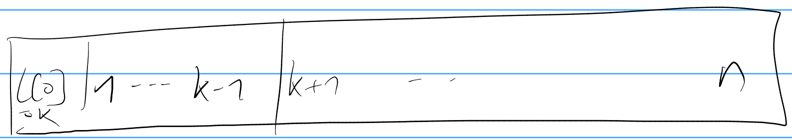
$\leq \sum_{i=1}^{k-1} \frac{n-k}{n-i}$

Conditioned on $L[0]=k$

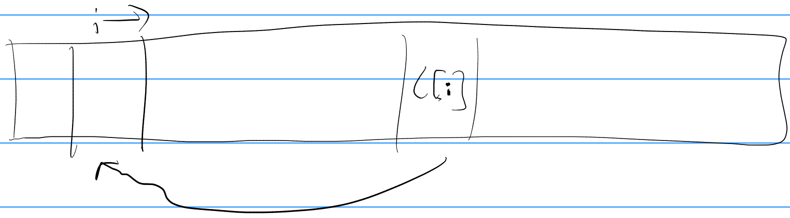
Total number (now $L[0]$ is arbitrary):

$$\mathbb{E}[\#\text{ swaps}] = \sum_{k=1}^{n} \Pr\left[L[0]=k\right] \mathbb{E}\left[\#\text{ swaps} \mid L[0]=k\right]$$

$$\leq \sum_{k=1}^{n} \frac{1}{n} \sum_{i=1}^{k-1} \frac{n-k}{n-i}$$



0 swaps



Clear bound on the # swaps: $k-1$ (for $L[0]=k$ fixed)
but we might do less.

$$\mathbb{E}[\#\text{Swaps}] = \sum_{k=1}^{n} \frac{1}{n} \mathbb{E}\left[\#\text{swaps} \mid L[0]=k\right]$$

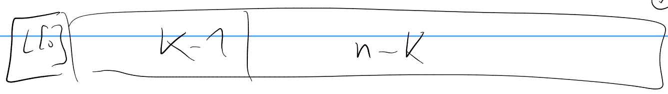
$$\leq \sum_{k=1}^{n} \frac{1}{n}(k-1) = \frac{1}{n}\left(\sum_{k=1}^{n} k-1\right) = \frac{1}{n}\sum_{k=0}^{n-1} k$$

$$\mathbb{E}[\#\text{Swaps}] \leq \frac{1}{n} \cdot \frac{n(n-1)}{2} \leq \frac{n-1}{2}$$

(2.)

C_n = average number of exchanges when sorting an array of size n.

$$C_n = \mathbb{E}[\#\text{ swaps}] + \sum_{k=1}^{n} \Pr[L[0]=k]\,\left(C_{k-1} + C_{n-k}\right)$$



$$C_n = \frac{n-1}{2} + \frac{1}{n}\sum_{k=1}^{n}\left(C_{k-1}+C_{n-k}\right)$$

$$= \frac{n-1}{2} + \frac{1}{n}\left(\sum_{k=0}^{n-1} C_k + \sum_{k=1}^{n} C_{n-k}\right)$$

$j = n-k$

$$= \sum_{j=0}^{n-1} C_j$$

$$\boxed{C_n = \frac{n-1}{2} + \frac{2}{n}\sum_{k=0}^{n-1} C_k} \quad (1)$$

make the sum "disappear" ?

$$C_{n-1} = \frac{n-2}{2} + \frac{2}{n-1}\sum_{k=0}^{n-2} C_k$$

$$\frac{n-1}{n} C_{n-1} = \frac{(n-1)(n-2)}{2n} + \frac{2}{n}\sum_{k=0}^{n-2} C_k \quad (2)$$

(1) − (2) $\quad C_n - \frac{n-1}{n}C_{n-1} = \frac{n-1}{2} - \frac{(n-1)(n-2)}{2n} + \frac{2}{n}C_{n-1}$

$$C_n = \Big(\underbrace{\frac{2}{n} + \frac{n-1}{n}}_{=\frac{n+1}{n}}\Big)C_{n-1} + \frac{n(n-1) - (n-1)(n-2)}{2n} = \frac{(n-(n-2))(n-1)}{2n}$$

$$= \frac{n-1}{n}$$

$$\boxed{C_n = \frac{n+1}{n}C_{n-1} + \frac{n-1}{n}} \quad (3) \qquad u_n = a\,u_{n-1} + f(n)$$

Put: $\boxed{u_n = \frac{C_n}{n+1}}$

$\frac{(3)}{n+1}$: $\quad u_n = u_{n-1} + \frac{n-1}{n(n+1)}$

$$u_n = \frac{n-1}{n(n+1)} + \frac{n-2}{(n-1)n} + u_{n-2}$$

$$= \text{-----}$$

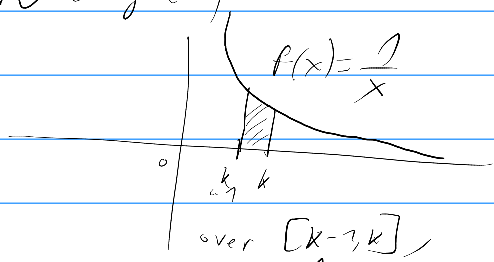
$$u_n = \sum_{k=0}^{n-1} \frac{n-k-1}{(n-k)(n-k+1)} \le$$

$\frac{a}{b\cdot c} \le \frac{1}{b}$ with $a \le c$

$$u_n \le \sum_{k=0}^{n-1} \frac{1}{n-k} \overset{j=n-k}{=} \sum_{j=1}^{n} \frac{1}{j}$$

Question: bound on $\sum_{k=1}^{n} \frac{1}{k}$? $\approx \log(n)$

2 methods: $$\sum_{k=2}^{n} \frac{1}{k} \le \sum_{k=2}^{n} \int_{k-1}^{k} f(x)\,dx \le \int_{\cancel{0}\,1}^{n-1} \frac{dx}{x} = \left[\ln(x)\right]_{\cancel{0}\,1}^{n-1}$$



over $[k-1, k]$, $f \ge \frac{1}{k}$

$\int_{k-1}^{k} f \ge \frac{1}{k}\,|\text{interval}|$, $k-(k-1) = 1$

$$\sum_{k=1}^{n} \frac{1}{k} = 1 + \sum_{k=2}^{n} \frac{1}{k} \le 1 + \left[\ln(x)\right]_{1}^{n-1} = 1 + \ln(n-1) - \ln(1) \le 1 + \ln(n-1) \approx \log(n).$$

$u_n = \frac{C_n}{n+1}$, $u_n \approx \log(n) = O(\log n)$

$$\boxed{C_n = O(n\log(n))}$$

on average, Quicksort makes $O(n\log(n))$ swaps.

Exercise 6: problem Hamiltonian cycle in a graph (a cycle that goes through every vertex exactly one time).

two parts:
- in NP
- NP-hard: take instance of Hamiltonian cycle, transform it into a knight problem.

$$\boxed{\text{Hamiltonian instance}} \xrightarrow[\text{polynomial time}]{\text{transform in}} \boxed{\text{Knight instance}}$$

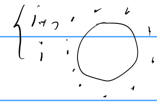
- if it has a solution $\longmapsto$ it has a solution
- if no solution $\longmapsto$ no solution

Problem is NP: is in NP

certificate

NP: we can verify/check a solution in polynomial time

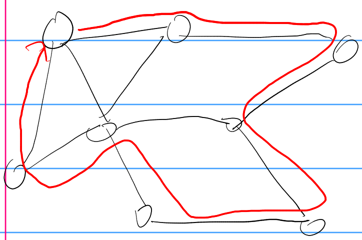
$\boxed{\text{certificate/solution}}$: a permutation / list of knight clockwise



$\boxed{\text{verify}}$: for every adjacent pair, check they are not enemies (go through the list of pairs)

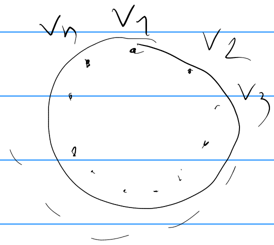
$\boxed{O(n^2)}$ $n = \#$ knights

NP-hard: take an instance of a hard problem (Hamiltonian) build an instance of our problem (knights)



Set of knights = Set of vertices
Set of enemies $= \{(u,v)$: no edge between u and $v\}$

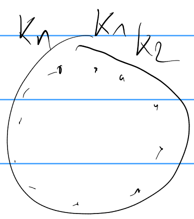
$\Rightarrow$ if there is a cycle C then we can order the knights for dinner: if $C = v_1, v_2, \dots, v_n, v_1$ then order



⚠ two directions are important

adjacent pairs: (v_i, v_{i+1}) and they are not enemies: because v_i has an edge to v_{i+1} (because in a cycle, must use edges)

$\Leftarrow$ if we can order the knights for dinner, we can find a H-cycle



build cycle $C = (k_1, k_2, \dots, k_n, k_1)$

- cycle: by def k_i, k_{i+1} not enemies therefore $\exists$ edge $k_i \to k_{i+1}$ so C consists of edges (and it goes back to k_1)
- Hamiltonian: length $n = \#$ vertices (ie every vertex appears exactly once).

Therefore, the knight-problem is at least as hard as the Hamiltonian cycle ", which is NP-hard. So it is NP-hard.

NP-hard
- SAT
- SUBSET-SUM
- TSP (travelling salesman problem)
- VERTEX-COVER
- 3-COLORING