Selection

How can we find the ith largest value

- in a sorted list?
- in an unsorted list?

Can we do better with an unsorted list than to sort it?

Assumption: Elements can be **ranked**.

Properties of Relationships

Partial Order: Given a set S and a binary operator R, R defines a partial order on S if R is:

- Antisymmetric: Whenever aRb and bRa, then $a = b$, for all $a, b \in \mathbf{S}$.
- Transitive: Whenever aRb and bRc, then aRc, for all $a, b, c \in \mathbf{S}$.

Think of a relationship as a set of tuples.

- A tuple is in the set (in the relation) iff the relation holds on that tuple.

Example: S is Integers, R is $<$.

Example: S is the power set of $\{1, 2, 3\}$, R is subset.

A partial order is also called a **poset**.

If every pair of elements in S is relatable by R, then we have a **linear order**.

General Model

For all of our problems on Selection and Sorting:

- The poset has a linear ordering. (Usually natural numbers and a relationship of $\leq$.)
- Cost measure is the number of 3-way element-element comparisons.

Selection problems:

- Find the max or min.
- Find the second largest.
- Find the median.
- Find the ith largest.
- Find several ranks simultaneously.

Finding the Maximum

```
int Find_max(int *L, int low, int high) {
  max = low;
  for(i=low+1; i<= high; i++)
    if(L[i] > L[max])
      max = i;
  return max;
}
```

What is the cost?

Is this optimal?

Proof of Lower Bound

Try #1:

- The winner must compare against all other elements, so there must be $n-1$ comparisons.

Try #2:

- Only the winner does not lose.
- There are $n-1$ losers.
- A single comparison generates (at most) one (new) loser.
- Therefore, there must be $n-1$ comparisons.

Alternative proof:

- To find the max, we must build a poset having one max and $n-1$ losers, starting from a poset of n singletons.
- We wish to connect the elements of the poset with the minimum number of links.
- This requires at least $n-1$ links.
- A comparison provides at most one new link.

Average Cost

What is the average cost for `Find_max`?

- Since it always does the same number of comparisons, clearly $n - 1$ comparisons.

How many assignments to `max` does it do?

Ignoring the actual values in L, there are $n!$ permutations for the input.

`Find_max` does an assignment on the ith iteration iff $L[i]$ is the biggest of the first i elements.

Since this event does happen, or does not happen:

- Given no information about distribution, the probability of an assignment after each comparison is 50%.

Average Number of Assignments

`Find_max` does an assignment on the ith iteration iff $L[i]$ is the biggest the first i elements.

Assuming all permutations are equally likely, the probability of this being true is $1/i$.

$$1 + \sum_{i=2}^{n} \frac{1}{i} \times 1 = \sum_{i=1}^{n} \frac{1}{i}.$$

This sum generates the nth **harmonic number**: $\mathcal{H}_n$.

Technique

Since $i \leq 2^{\lceil \log i \rceil}$, $1/i \geq 1/2^{\lceil \log i \rceil}$.

Thus, if $n = 2^k$

$$\begin{aligned}
\mathcal{H}_{2^k} &= 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{2^k} \\
&\geq 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} \\
&\qquad + \ldots + \frac{1}{2^k} \\
&= 1 + \frac{1}{2} + \frac{2}{4} + \frac{4}{8} + \ldots \frac{2^{k-1}}{2^k} \\
&= 1 + \frac{k}{2}.
\end{aligned}$$

Using similar logic, $\mathcal{H}_{2^k} \leq k + \frac{1}{2^k}$.

Thus, $\mathcal{H}_n = \Theta(\log n)$.

More exactly, $\mathcal{H}_n$ is close to $\ln n$.

Variance

How "reliable" is the average?

- How much will a given run of the program deviate from the average?

Variance: For runs of the program, average square of differences.

Standard deviation: Square root of variance.

From Čebyšev's Inequality, 75% of the observations fall within 2 standard deviations of the average.

For `Find_max`, the variance is

$$\mathcal{H}_n - \frac{\pi^2}{6} = \ln n - \frac{\pi^2}{6}$$

The standard deviation is thus about $\sqrt{\ln n}$.

- So, 75% of the observations are between $\ln n - 2\sqrt{\ln n}$ and $\ln n + 2\sqrt{\ln n}$.
- Is this a narrow spread or a wide spread?