

Design and Analysis of Algorithms

L10b: Bonus Exercises

Dr. Ram P Rustagi
Sem IV (2019-H1)
Dept of CSE, KSIT/KSSEM
rprustagi@ksit.edu.in

Resources

- <https://cs.stackexchange.com/questions/10960/sort-array-of-5-integers-with-a-max-of-7-compares>

Sort K elements

- Minimum comparisons required
- With K elements, possible permutations $K!$
- Only 1 out $K!$ is correct sorted order.
- Using binary partition, the minimum comparison required
 $\log_2 K!$
- Example:
 - $K=4$, Min comparisons : $\log_2 24=5$
 - Develop a decision tree to sort 4 elems in 5 comparisons
 - $K=5$, Min comparisons : $\log_2 120=7$
 - Develop a decision tree to sort 5 elems in 7 comparisons

Sort 4 elements

- Task: sort 4 elements in precisely 5 comparisons
 - Input: 4 elements (or their 24 permutations)
 - a, b, c, d .
 - output: pick 1 correct out of 24 permutations
- Methodology
 - Ensure each comparison reduces the set by half
 - In i^{th} comparison, the permutation sets size be
 - set size $\leq 2^{5-i}$
 - Thus, permutation set size should decrease with each comparison to be $\leq 16, 8, 4, 2, 1$
 - Devise a method where permutation set size decreases from 24 to 12, 6, 3, 2, 1

Sort 4 elements...

- C1: Compare a and b . For generality, assume $a < b$
- Given this condition, possible permutations = 24
 - c can be placed 3 ways
 - before a , i.e. $(c \ a \ b)$
 - between a and b $(a \ c \ b)$
 - after b . $(a \ b \ c)$
 - for each possible placement of c , d can be placed in 4 possible ways
 - $dcab, \ cdab, \ cadb, \ cabd$
 - $dacb, \ adcb, \ acdb, \ acbd$
 - $dabc, \ adbc, \ abdc, \ abcd$
 - Total permutations: $3 * 4 = 12 \ (< 16)$.
 - This satisfies our division criteria.

Sort 4 elements...

- C2: Possible comparisons
 - compare c and d , or
 - c with a , or (even d can be taken in place of c)
 - c with b .
- Consider $C2_1$: compare c with d .
 - For generality, assume $c < d$.
 - This partitions the permutation set from 12 to 6.
 - $cdab, cadb, cabd$
 - $acdb, acbd$
 - $abcd$
 - Total permutations: 6 (< 8) .
 - Follows constraints $a < b$, and $c < d$
 - This satisfies our division criteria.

Sort 4 elements...

- Permutation set after 2 comparisons ($a < b, c < d$)
 - cdab, cadb, cabd, acdb, acbd, abcd
- C3: it should divide the set into half i.e. size of 3
 - Comparing c and b, gives following division
 - ($c < b$) : cdab, cadb, cabd, acdb, acbd
 - ($b < c$) : abcd
 - Division divides into 5 and 1 and $5 > 3$. So this comparison will not work.
 - Similarly, comparing a and d divides the set into subset of size 5 and size 1.
 - Comparing b and d gives equal division of 3, and 3.
 - ($b < d$) : cabd, acbd, abcd
 - ($d < b$) : cdab, cadb, acdb

Sort 4 elements...

- Thus, **C3**: compare b and d to get following.
 - $C_{3a} : (b < d) : cabd, acbd, abcd$
 - $C_{3b} : (d < b) : cdab, cadb, acdb$
 - Both sets are size 3 (< 4) and thus works fine.
- $C_{3a} : a < b, c < d, b < d \Rightarrow a < b < d, c < d$
 - We don't about order of $(a < b)$ and c .
- **C4_{3a}**: Compare a and c .
 - $C_{4a} : a < c \Rightarrow acbd, abcd$
 - $C_{4b} : c < a \Rightarrow cabd$. (Done)
- **C5_{3a}**: Compare b and c .
 - **C5a**: $b < c \Rightarrow abcd$ (Done)
 - **C5a**: $c < b \Rightarrow acbd$ (Done)
- We can similarly complete C_{3b} in 5 comparisons

Bonus exercise: Sort 5 elements

- Task: sort 5 elements in precisely 7 comparisons
 - Input: 5 elements (or their 120 permutations)
 - a, b, c, d, e .
 - output: pick 1 correct out of 120 permutations
- Methodology
 - Ensure each comparison reduces the set by half
 - In i^{th} comparison, the permutation sets size be
 - set size $\leq 2^{7-i}$
 - Thus, permutation set size should decrease with each comparison to be $\leq 64, 32, 16, 8, 4, 2, 1$
 - Devise a method where permutation set size decreases from 120 to $60, 30, 15, 8, 4, 2, 1$

Sort 5 elements...

- C1: Compare a and b. For generality, assume $a < b$
- Given this condition, possible permutations = 60
 - c can be placed 3 ways
 - (c a b), (a c b) or (a b c)
 - for each possible placement of c, d can be placed in 4 possible ways
 - dcab, cdab, cadb, cabd
 - dacb, adcb, acdb, acbd
 - dabc, adbc, abdc, abcd
 - For each of these 12, e can be placed 5 ways.
 - Total permutations: $3 * 4 * 5 = 60$ (< 64) .
 - This satisfies our division criteria.

Sort 5 elements...

- C2: compare c and d
- Consider for generality, assume $c < d$.
 - This partitions the permutation set from 60 to 30.
 - 30 is < 32 , satisfies the division criteria
- C3: Proceed further in this way to have a set division from 30 to 15
- C4: Set division from 15 to 8
- C5: set division from 8 to 4
- C6: set division from 4 to 2
- C7: set division from 2 to 1 (get the sorted set)
- Please work out the steps!!!
 - Write the sorting program to see the results.