CH 7 QUICKSORT
CH. 5 & APPOX C.

PROBLEM w/ MERGESORT is SPACE is $\Theta(n)$.

QUICKSORT idea is choose a pivot & place all < pivot on LHS & all ≥ pivot on RHS.
PIVOT will be in final place. then do same to each side

EX: suppose we use last element as pivot:

1	2	3	4	5	6
2	5	7	1	4	3
					pivot

keep index of last of LHS & 1^{st} UNK.

2 < 3 so last of LHS is 1.
5 > 3
7 > 3
so

	RHS				
2	5	7	1	4	3
last LHS			1^{st} UNK.		pivot

1 < 3 swap with 5. ⟹

		RHS			
2	1	7	5	4	3
	last LHS			1^{st} UNK	pivot

4 > 3 so it stays as RHS.

finally swap 3 w/ last LHS + 1 index.

2	1	$\boxed{3}$	5	4	7

This is called <u>partition</u> & it is done <u>in place</u>

then do partition on each side (like Mergesort).

What does the PARTITION FUNCTION NEED AS PARAMS.?

LIST, START & END INDEXES

WHAT OTHER VARS WILL IT NEED?

PIVOT, LASTLHS, FIRSTUNK

WILL BE CALLED LIKE:

```
PIVOT ← PARTITION(LIST, START, END)
```

```
PARTITION(LIST, START, END)
PIVOT ← END
LASTLHS ← START - 1
FIRSTUNK ← START
while FIRSTUNK < END
    if LIST[FIRSTUNK] < LIST[PIVOT]
        LASTLHS++
        SWAP(LIST, FIRSTUNK, LASTLHS)
    FIRSTUNK++
SWAP(LIST, PIVOT, LASTLHS+1)
RETURN. LASTLHS+1.
```

```
QUICKSORT(LIST, START, END)
IF START < END
    PIVOT ← PARTITION(LIST, START, END)
    QUICKSORT(LIST, START, PIVOT - 1)
       "     (  "  , PIVOT+1, END)
```

When we have an algorithm what do we usually like to analyze?

Space &

Runtime

- Best Case
- Worst case

HOW LONG DOES PARTITION TAKE?

loop iterates n times when $n = \text{END} - \text{START} + 1$

$\Rightarrow \Theta(n)$

For Best Case

do fewest calls to QS. $\Rightarrow$ cut in HALF all the time:

$$n$$

$$\frac{n}{2} \quad \frac{n}{2}$$

$\leftarrow$ note each is off by 1.

$\vdots$

$1 \quad 1 \quad 1 \quad \ldots \quad 1$

this should look familiar.

BEST CASE: $T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n)$

$a = 2, b = 2 \quad f(n) = \Theta(n) \quad n^{\log_b a} = n = \Theta(n)$

CASE 2: $T(n) = \Theta(n \lg n)$

WORST CASE: ideas?

example

1	2	3	4	5	6	7	8	9	10

	TIME
n	n
	+
$n-1$	$n-1$
	+
$n-2$	$n-2$
	+
⋮	⋮
	+
1	1

$$\sum_{i=1}^{n} i \stackrel{?}{=} \frac{n(n+1)}{2}$$

1ST QS call, 2ND QS call,

$$T(n) = T(n-1) + T(0) + \Theta(n)$$

Worst case $\Theta(n^2)$

An already sorted list is Worst Case $\Rightarrow \Theta(n^2)$

So OVERALL for QS: $O(n^2)$, $\Omega(n\lg n)$

- WE would like no particular input to cause worst case n^2
  - → don't want an adversary to cause n^2 performance ⇒ slow

random

Replace 1st line of PARTITION: RANDIDX ← RAND(START, END)
SWAP(LIST, END, RANDIDX)
PIVOT ← END

this is now a randomized algorithm.

are we closer to $n \lg n$ or n^2?

NEED to know some probability concepts to answer this.

PROBABILITY

flip a coin – if both sides equally likely? $\Longrightarrow$

Likelihood of HEADS = 50%

If flip it a million times expect 50% heads

Defns:

$\boxed{\text{SAMPLE SPACE}}$ S is set of all possible outcomes

e.g. $S_{coin} = \{H, T\}$ for 1 flip.

We have a probability weight associated w/ each outcome.

1. each weight $w_i \geq 0$
2. $\sum_i w_i = 1$

e.g. flip 3 times:

$S = \{HHH, HHT, HTH, HTT, THH, THT, TTH, TTT\}$

each w/ $w = \frac{1}{8}$

What is the PROB. @ least 1 H occurs? $\longleftarrow$ ex. of AN $\boxed{\text{EVENT}}$

there is no weight assoc. w/ this
if there were, it would be an **ELEMENTARY EVENT** aka **OUTCOME** of our experiment

An **EVENT** is a SUBSET OF **SAMPLE SPACE**

if E is an EVENT

$$P(E) = \text{add up the weights in SS that satisfy } E$$

$$= \sum_{x \in E} p(x)$$

$$= 7/8 \text{ in our example.}$$

A **PROBABILITY FUNCTION** on a SAMPLE SPACE S satisfies.

1. $P(A) \geq 0 \quad \forall A \subseteq S$
2. if A and B are disjoint EVENTS then
   $P(A \cup B) = P(A) + P(B)$
3. $P(S) = 1$

if E is an EVENT $\bar{E}$ is **complement** of E, $\bar{E} = S - E$

$$P(E) = 1 - P(E)$$

$$\text{b/c.} \quad P(S) = P(E \cup \bar{E}) = P(E) + P(\bar{E}) = 1$$