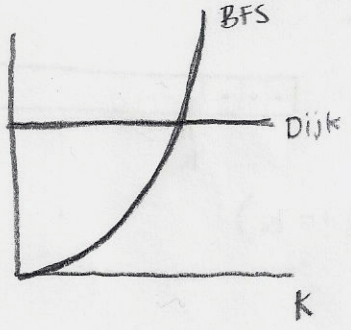
CS157 Lecture #4

↳ connect 2 actors, where actors are linked if they've been in the same movie

Q: How do you solve "6 Degrees of Kevin Bacon" given the data of all movies ever?

- graph that connects all the actors
  - ↳ let's do a breadth first search! except this isn't a tree...
  - ↳ let's do Dijkstra's! (so we can mark nodes we've visited)
    - ↳ = $O(|V|+|E|)$ time
      $O(V)$ memory
  - ↳ BFS = $O(n^k)$ time, n = avg # movies an actor's in, k is the max length path you want
    = $O(k)$ memory

* BFS returns short paths quickly, but takes a long time for longer time
* Dijkstra's takes longer than BFS for short paths, but is better over time

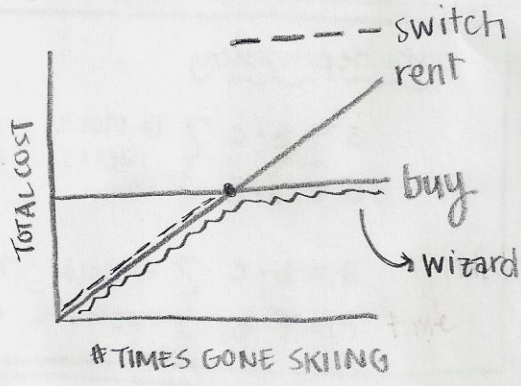


WHICH IS BETTER?

★ A RELATED PROBLEM

You go skiing. It costs $50 to rent skis each time you go, and $500 to buy them. What do you do?

↳ ideas: ① hedge your bets: just buy them
② rent until you've spent a certain amount, and then buy them
↳ where to switch? intersection.



★ COMPETITIVE ANALYSIS

: you're competing with the optimal solution (controlled by a wizard who knows the future). You can't match it (because you don't know the future), but can you be competitive with it?

↓

In the ski-rental problem, you're never worse than 2-times the cost of the wizard solution, which makes the switching solution 2-competitive.

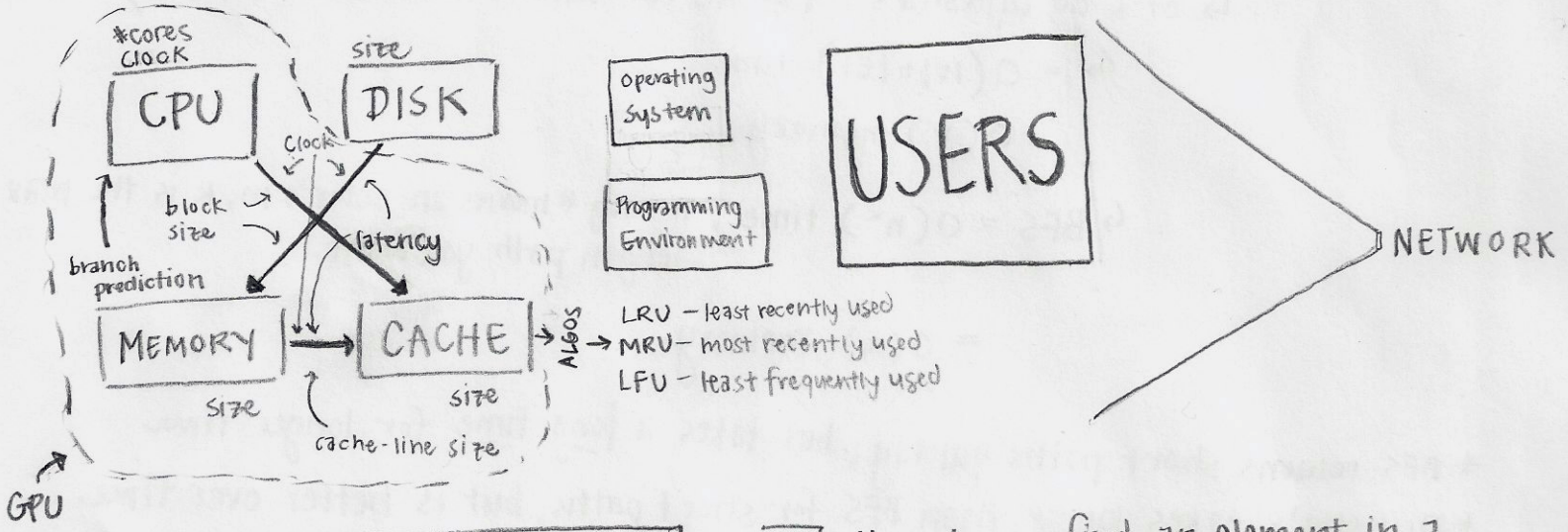
EDIT DISTANCE

↳ 3 GHz computer, 2 strings of length 10,000 (filling in 10,000 × 10,000 table) $\approx 10^8$

- How many clock cycles does it take to fill in 1 entry of the table?

$$= \frac{3 \cdot 10^9 \cdot 2}{10^8} = 60 \text{ clock cycles}$$

MAP OF THE COMPUTER



*data dependency

$a = b \cdot c$ } 16 clock cycles — need to wait for first to finish
$d = a \cdot e$ }

$a = b \cdot c$ } 9 clock cycles — 2nd starts 1 after first
$d = f \cdot e$ }

Q: How do you find an element in a sorted list?

A1: Linear search

↓

```
while (array[i] != k)
    i++
```

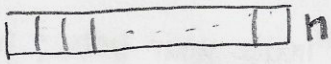
↓

each iteration will take ≈ 4 cycles if branch prediction works. If it doesn't, it will take ≈ 25 cycles

* if searching long away, you're going to bet you have to keep looking → you're going to guess wrong once

⇒ total cost: $k+b \longrightarrow \frac{n}{2} + b$, b = depth of pipeline

A2: Binary Search



* takes $\log_2 n$ iterations to find elem.
* each iteration costs $\frac{b}{2} + 1$, since we'll be wrong about ½ the time

⇒ total cost = $(\frac{b}{2} + 1) \log_2 n$