# DSC 40B Lecture 6 : Average Cases



# Time Taken, Typically

- Best case and worst case can be misleading.
  - Depend on a single good/bad input.
- How much time is taken, typically?
- **Idea**: compute the average time taken over *all possible* inputs.



• The **expected value** of a random variable *X* is:

$$\sum_X x \cdot P(X = x)$$

winnings probability
\$ 0
\$ 1
\$ 10
\$ 10
\$ 50
2%

**Expected winnings:** 



ullet The **expected value** of a random variable X is:

$$\sum_X x \cdot P(X = x)$$

winnings probability \$ 0 50% \$ 1 30% \$ 10 18% \$ 50 2%

#### **Expected winnings:**



• The **expected value** of a random variable *X* is:

$$\sum_X x \cdot P(X = x)$$

winnings **Expected winnings:** probability 50% 30%  $$0 \times .5 + $1 \times .3$ \$ 10 \$ 50 18%

2%



• The **expected value** of a random variable *X* is:

$$\sum_X x \cdot P(X = x)$$

winnings	probability	Expected winnings:
\$0	50%	
\$1	30%	\$0 x .5 + \$1 x .3 + \$10 x .18 +
\$ 10	18%	
\$ 50	2%	



\$ 50

#### Recall: The Expectation

2%

• The **expected value** of a random variable *X* is:

$$\sum_X x \cdot P(X = x)$$

 winnings
 probability
 Expected winnings:

 \$ 0
 50%

 \$ 1
 30%

 \$ 10
 \$0 x .5 + \$1 x .3 + \$10 x .18 + \$50 x .02



ullet The **expected value** of a random variable X is:

$$\sum_X x \cdot P(X = x)$$

winnings probability

\$ 0 50% \$ 1 30% \$ 10 18% \$ 50 2%

#### Expected winnings:

$$$0 \times .5 + $1 \times .3 + $10 \times .18 + $50 \times .02 = $3.10$$



#### Average Case

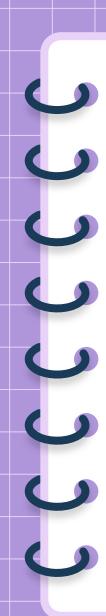
We'll compute the expected time over all cases:

$$T_{\text{avg}}(n) = \sum_{\text{case} \in \text{all cases}} P(\text{case}) \cdot T(\text{case})$$

Probability of the case

Time of the case

• Called the average case time complexity.



#### Strategy for Finding Average Case

- **Step 0**: Make *assumption* about distribution of inputs.
- **Step 1**: Determine the *possible* cases.
- **Step 2**: Determine the *probability* of each case.
- **Step 3**: Determine the *time* taken for each case.
- **Step 4**: Compute the expected time (average).

#### Example: Linear Search

Best? Worst? Average?

```
def linear_search(arr, t):
    for i, x in enumerate(arr):
        if x == t:
        return i
    return None
```



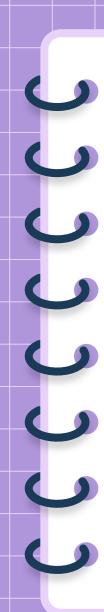
# Example: Linear Search

 What is the average case time complexity of linear search?



## Step 0: Assume input distribution

- We must assume something about the input.
- Example:
  - Target must be in array,
  - o equally-likely to be any element,
  - o no duplicates.
- This is *usually* given to you.



#### Step 1: Determine the Cases

**Example**: linear search.

- Case 1: target is first element
- Case 2: target is second element

- Case *n*: target is *n*th element
- Case n + 1: target is not in array (but not needed due to assumptions)



## Step 2: Case Probabilities

- What is the probability that we see each case?
  - Example: what is the probability that the target is the kth element?
- This is where we use assumptions from Step 0.



#### Example

- **Assume**: target is in the array exactly once, equally-likely to be any element.
- Each case has probability 1/n.
  - Uniform probability



#### Step 3: Case Times

- Determine **time taken** in each case.
- **Example**: linear search.
  - $\circ$  Let's say it takes time c per iteration.

Case 1: time c #some constant

Case 2: time 2c

:

Case i: time  $c \cdot i$ 

:

Case n: time  $c \cdot n$ 

#### Step 3: Case Times

- Determine **time taken** in each case.
- **Example**: linear search.
  - $\circ$  Let's say it takes time c per iteration.

Case 1: time c #some constant

Case 2: time 2c

:

 $T(case i) = c \cdot i$ 

Case i: time  $c \cdot i$ 

:

Case n: time  $c \cdot n$ 



$$T_{\text{avg}}(n) = \sum_{i=1}^{n} P(\text{case } i) \cdot T(\text{case } i)$$



$$T_{\text{avg}}(n) = \sum_{i=1}^{n} P(\text{case } i) \cdot T(\text{case } i)$$

$$\sum_{i=1}^{n} \frac{1}{n} \cdot ic$$



$$T_{\text{avg}}(n) = \sum_{i=1}^{n} P(\text{case } i) \cdot T(\text{case } i)$$

$$\sum_{i=1}^{n} \frac{1}{n} \cdot ic$$

$$\frac{c}{n}\sum_{i=1}^{n}i$$

$$T_{\text{avg}}(n) = \sum_{i=1}^{n} P(\text{case } i) \cdot T(\text{case } i)$$

$$\sum_{i=1}^{n} \frac{1}{n} \cdot ic$$

$$\frac{c}{n} \sum_{i=1}^{n} i = \frac{c}{n} (1 + 2 + \dots + n)$$

$$T_{\text{avg}}(n) = \sum_{i=1}^{n} P(\text{case } i) \cdot T(\text{case } i)$$

$$\sum_{i=1}^{n} \frac{1}{n} \cdot ic$$

$$\frac{c}{n} \sum_{i=1}^{n} i = \frac{c}{n} (1 + 2 + \dots + n) = \frac{c}{n} \cdot \frac{n(n+1)}{2}$$

# ن

$$T_{\text{avg}}(n) = \sum_{i=1}^{n} P(\text{case } i) \cdot T(\text{case } i)$$

$$\sum_{i=1}^{n} \frac{1}{n} \cdot ic$$

$$\frac{c}{n} \sum_{i=1}^{n} i = \frac{c}{n} (1 + 2 + \dots + n) = \frac{c}{n} \cdot \frac{n(n+1)}{2}$$

$$\frac{c(n+1)}{2} = \Theta(n)$$



#### Average Case Time Complexity

- The average case time complexity of linear search is  $\Theta(n)$ .
  - Output these assumptions on the input!



#### Note

- Worst case time complexity is still useful.
- Easier to calculate.
- Often same as average case (but not always!)
- Sometimes worst case is very important.
  - Real time applications, time complexity attacks

# Thank you!

Do you have any questions?

CampusWire!