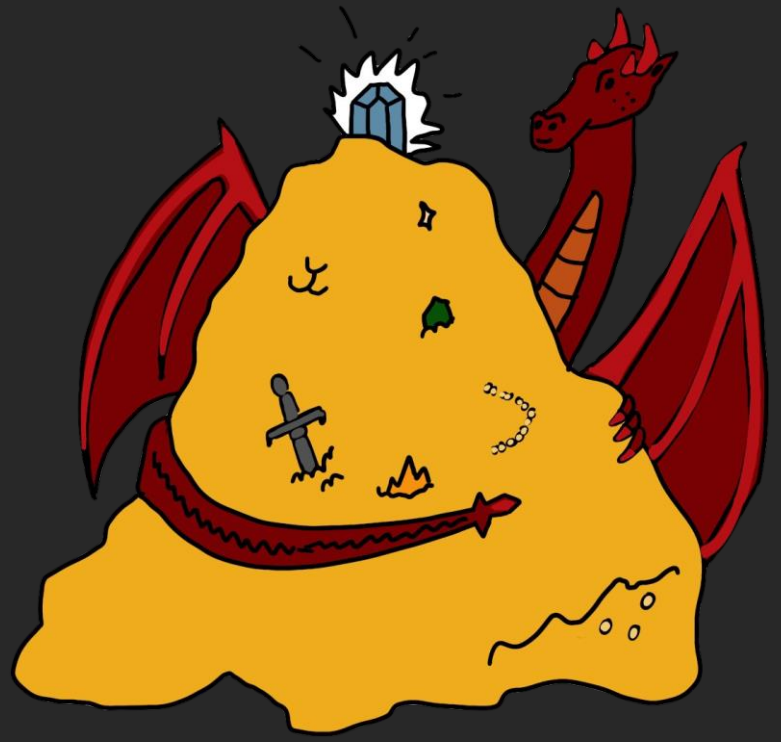
CSCI 3104
Lecture 8: Greedy Algorithms

Caleb Escobedo

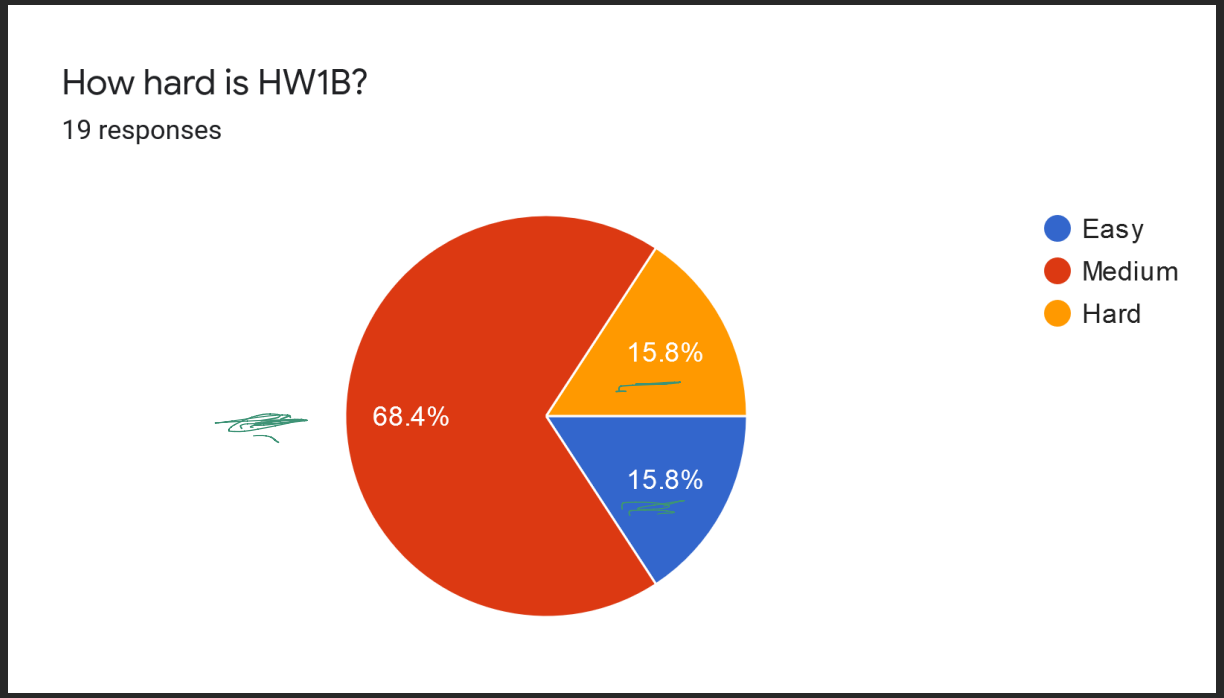
Caleb.Escobedo@colorado.edu

Lecture Outline

- Poll review/Administrative
- Greedy Algorithms
  - Introduction
  - Example
  - Structure
- Huffman Encoding
  - Introduction
  - History
  - Terminology
  - Example
- Homework examples



Poll Review!

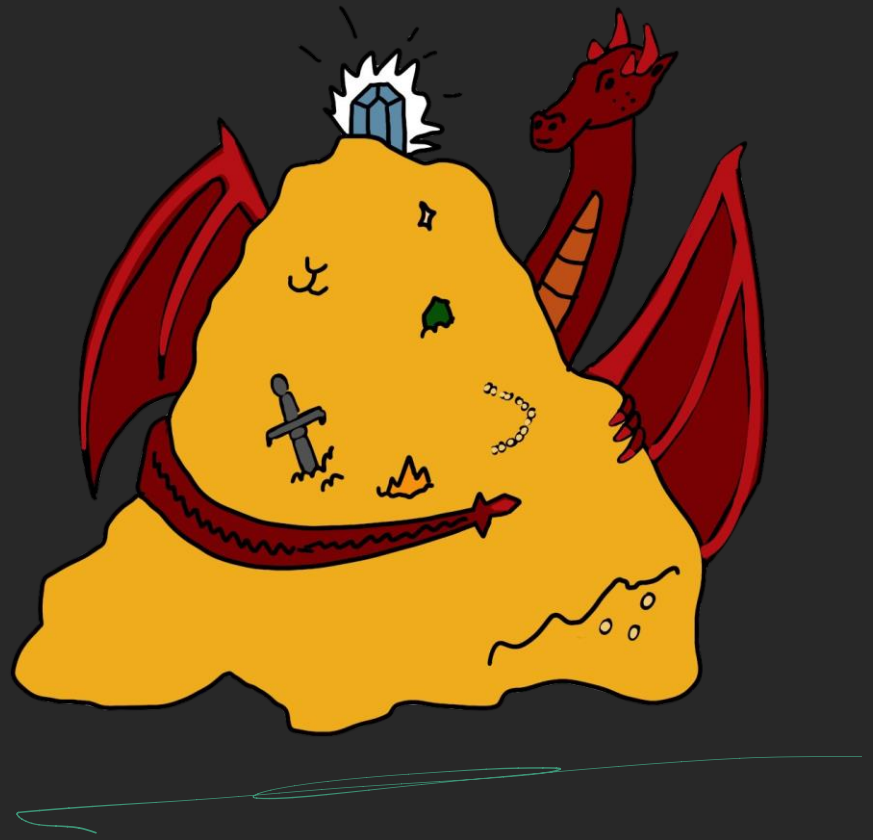


Administrative

- Homework 2A due on Tuesday
- Midterm will be released Friday the 26th and due on Monday the 29th
- Will we release homework solutions? Yes
- What do you need to show on the homework?
  - We are trying to be more explicit! Ask on Piazza if needed

Lecture Outline

- ~~Poll review/Administrative~~
- Greedy Algorithms
  - Introduction
  - Example
  - Structure
- Huffman Encoding
  - Introduction
  - History
  - Terminology
  - Example
- Homework examples



Greedy Algorithms: Introduction

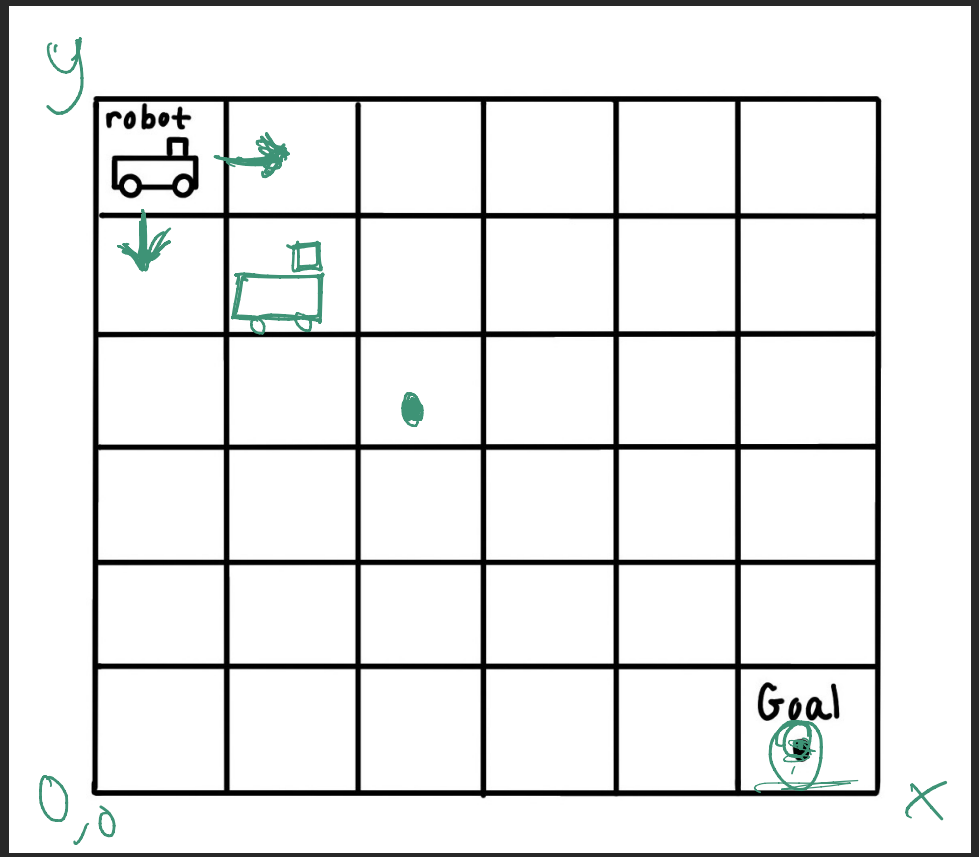
- Breaks a problem down into a sequence of simple steps each of which is chosen such that a measure of "quality" with the intermediate solutions increases a maximal amount of reward.
- The equation below could be used as a measure of quality

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

- When does this work?
  - Space is smooth
  - No impassable regions

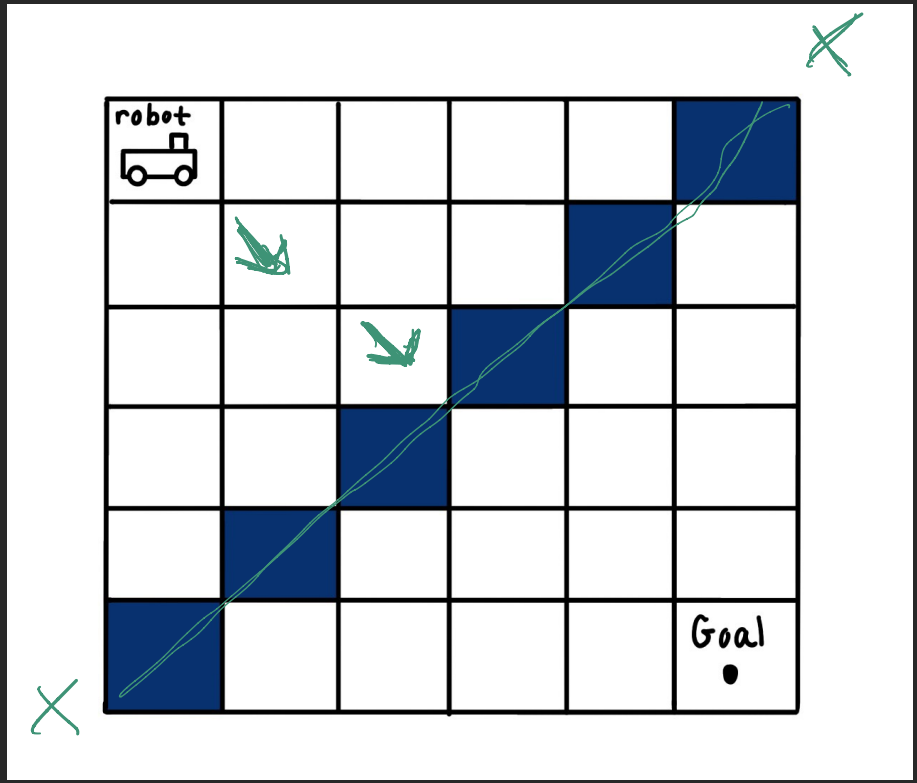
Greedy Algorithms: Example 1

- What is the optimal move for our robot to take at this step?
  - Y += 0, 1, or -1
  - X += 0, 1, or -1



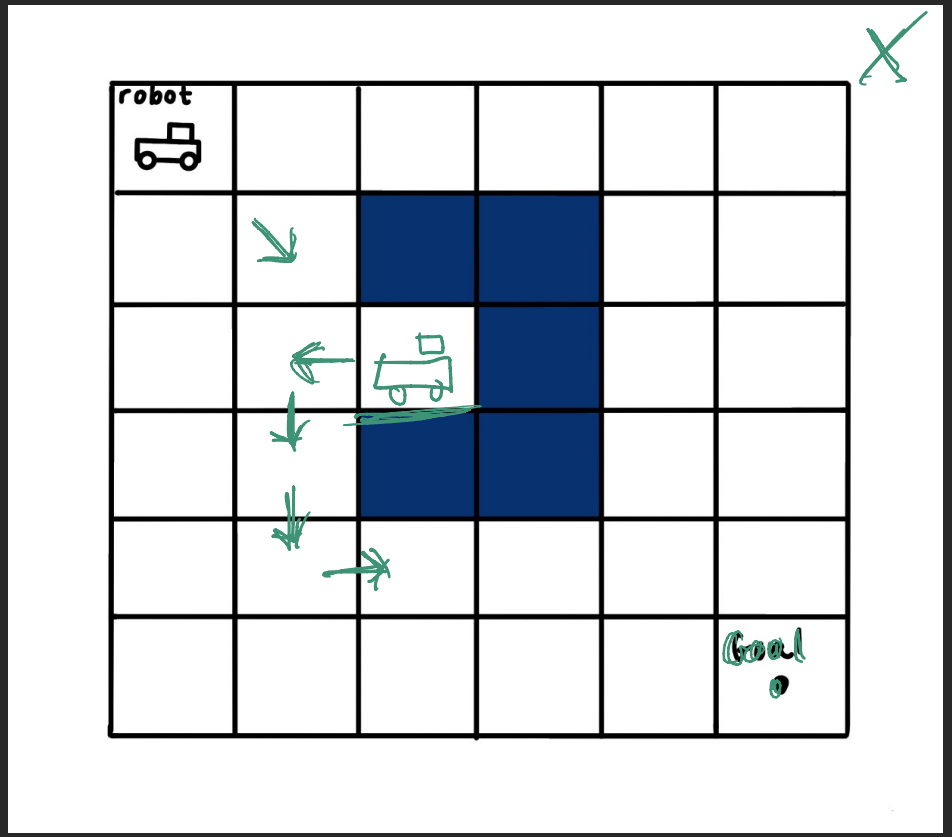
Greedy Algorithms: Example 2

- What is the optimal move for our robot to take at this step?
  - Y += 0, 1, or -1
  - X += 0, 1, or -1



Greedy Algorithms: Example 3

- What is the optimal move for our robot to take at this step?
  - Y += 0, 1, or -1
  - X += 0, 1, or -1

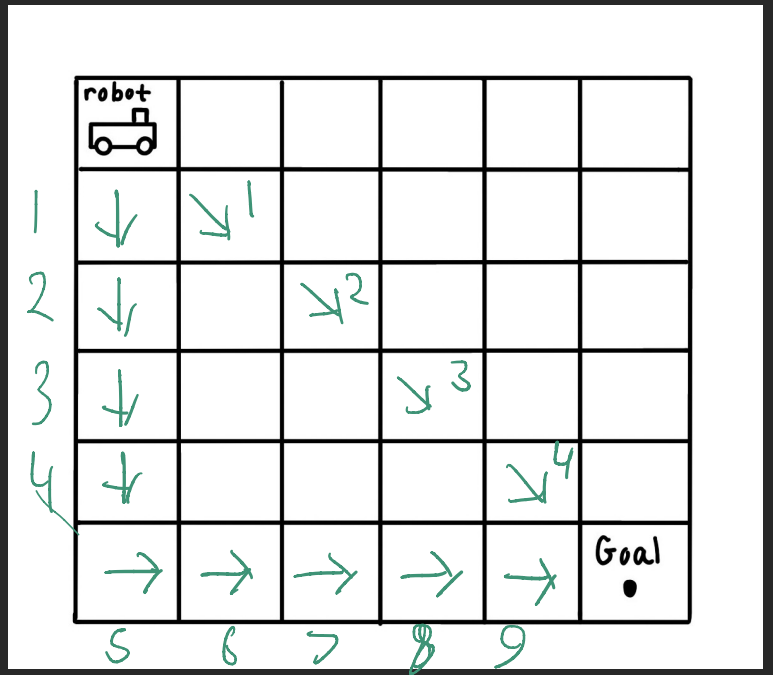


Greedy Algorithms: Structure

- A greedy algorithm always chooses the incremental option that yields the largest improvement in the intermediate solution's score.
- To prove a greedy algorithm is correct we can show that is has both the optimal substructure property and the greedy choice property
- If we can prove that a problem has the following 5 properties (next 5 slides), then there must exist a greedy algorithm for solving it

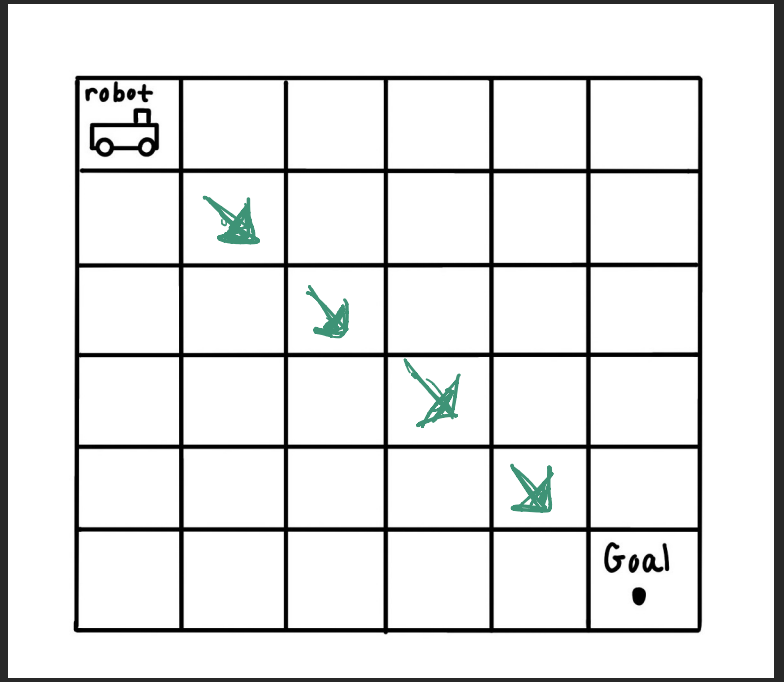
Greedy Algorithms: Structure 1

1) Every solution to the problem can be assigned or translated in a numerical value or score. Let x be a candidate solution and let score(x) be the value assigned to that solution



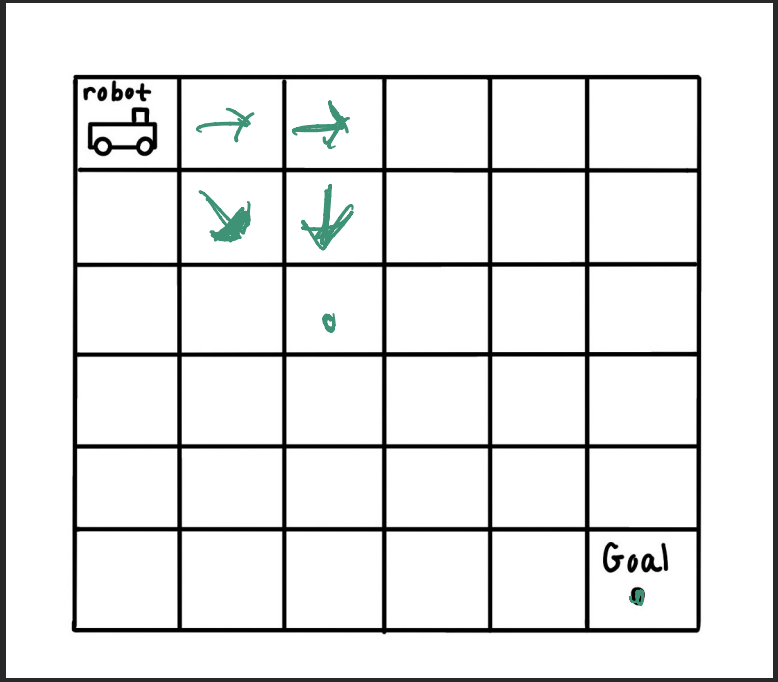
Greedy Algorithms: Structure 2

2) There exists a “best” (optimal) solution, with the highest (or lowest) score among all solutions



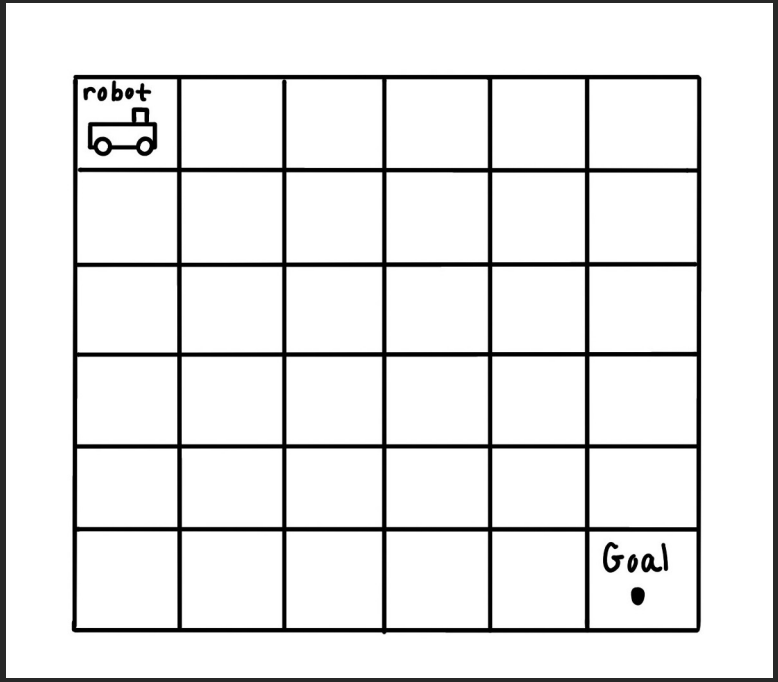
Greedy Algorithms: Structure 3

3) Optimal substructure property: the optimal solution contains optimal solutions to all its subproblems



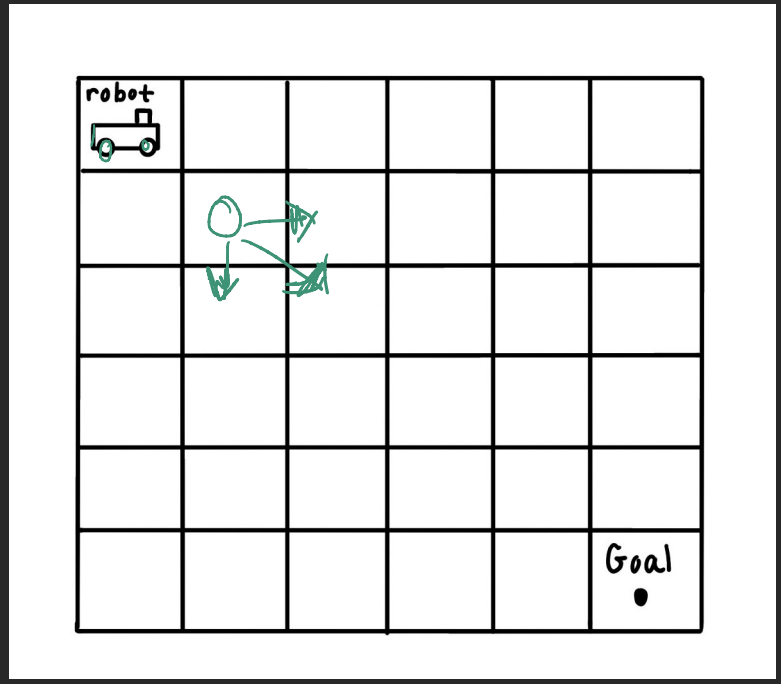
Greedy Algorithms: Structure 4

4) Greedy choice property: a solution can be constructed incrementally (and a partial solution assigned a score) without reference to future decisions, past decisions, or the set of possible solutions to the problem.



Greedy Algorithms: Structure 5

5) At each intermediate step in constructing or finding a solution, there are a set of options for which piece to add next.

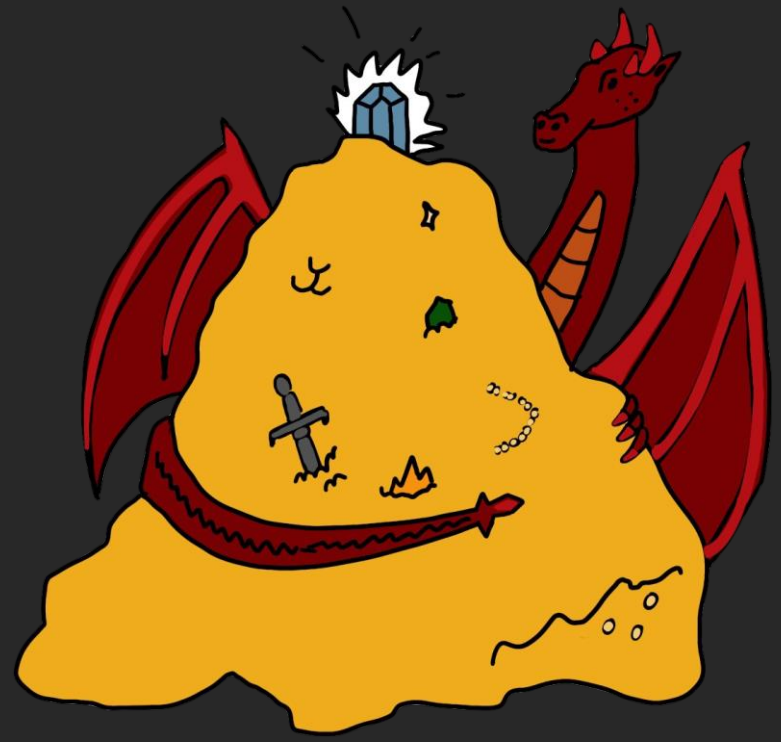


Greedy Algorithms: Structure

1) Every solution to the problem can be assigned or translated in a numerical value or score. Let x be a candidate solution and let score(x) be the value assigned to that solution
2) There exists a “best” (optimal) solution, with the highest (or lowest) score among all solutions
3) Optimal substructure property: the optimal solution contains optimal solutions to all its subproblems
4) Greedy choice property: a solution can be constructed incrementally (and a partial solution assigned a score) without reference to future decisions, past decisions, or the set of possible solutions to the problem.
5) At each intermediate step in constructing or finding a solution, there are a set of options for which piece to add next

Lecture Outline

- ~~Poll review/Administrative~~
- ~~Greedy Algorithms~~
  - ~~Introduction~~
  - ~~Example~~
  - ~~Structure~~
- Huffman Encoding
  - Introduction
  - History
  - Terminology
  - Example
- Homework examples



Huffman: Introduction

Goal

- Input: “A message or any data that you want to compress, here we have a string”
- Output: the most compressed version of that data without any loss of information
  - **.zip, .tgz**
    - *Lossless* compression
  - Not **.jpg**
    - *Lossy* compression

Huffman: History

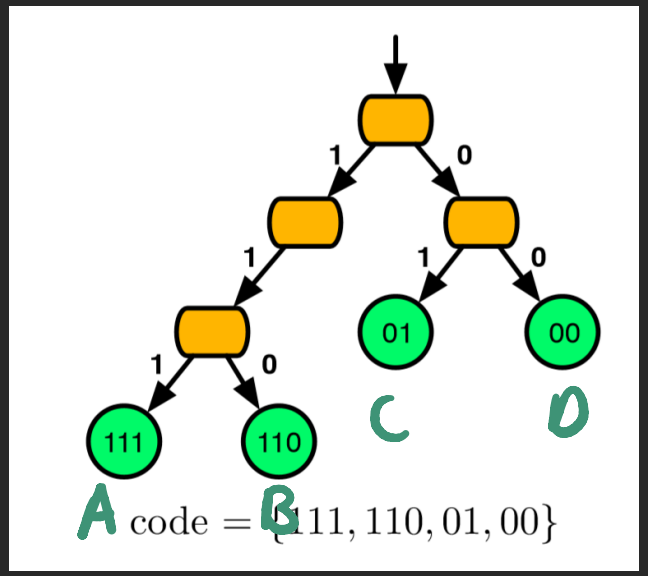
- David A. Huffman
  - PhD student at MIT
  - Taking a course on Information Theory by Robert M. Fano
  - Two choices for final
    - 1) Final exam
    - 2) Term paper -> find the most efficient binary code
      - Fano was really working on this research and did not have a solution
      - Huffman found and proved that his solution was optimal

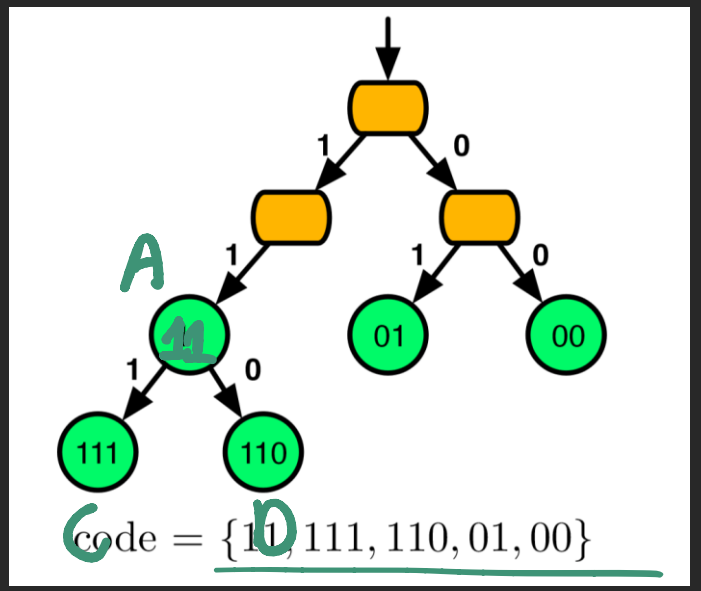
Huffman: Terminology

- Let Σ be the input alphabet with $|\Sigma| = n$ distinct “symbols” that can be used to construct a message
- The output of our algorithm is a mapping g of words from Σ to a binary alphabet Γ
- The encoding will be *prefix-free* and the encoded message will be of minimal size

Huffman: Terminology

- Prefix-free {111, 01, 110, 00}
- Not prefix-free {11, 111, 01, 110, 00}.



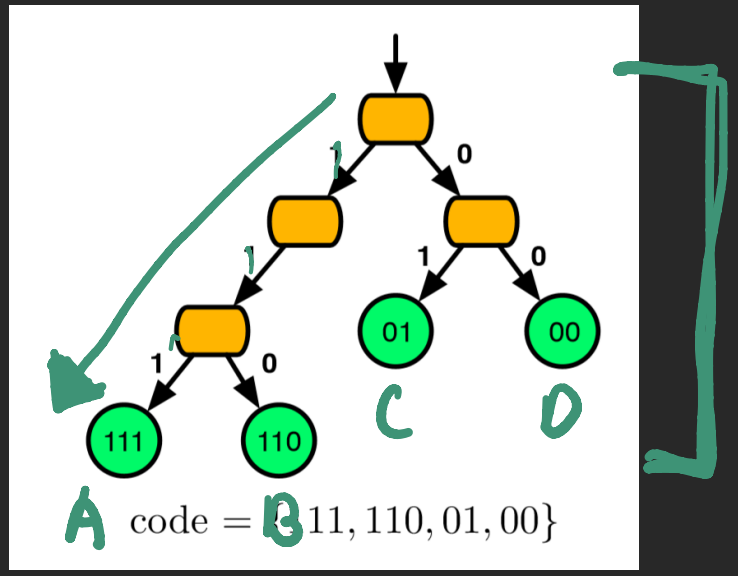


Huffman: Cost of Coding

"DDDA"

"ABCD" → 1111100100

"AAAD"



frequency

$$\text{cost}(x) = \sum_{i=1}^{n} f_i \cdot \text{depth}(i)$$

Huffman: Example

- Huffman developed a greedy algorithm that minimized the function

$$\text{cost}(x) = \sum_{i=1}^{n} f_i \cdot \text{depth}(i)$$

- Calculate f_i for each word in x; construct a histogram

"A A A B B C"

A:3, B:2, C:1

Huffman: Example

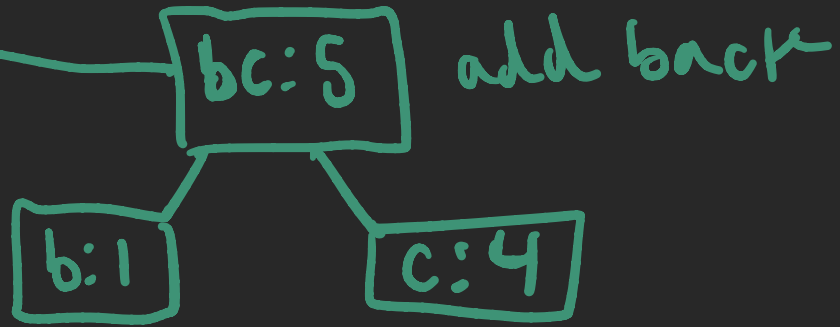
1) We reduced some message down to a set of symbols and their frequencies

a:6	b:1	c:4	d:4	e:7

2) Create a priority queue of this data with frequency determining the order

b:1, c:4, d:4, a:6, e:7

3) Deque 2 of the elements and merge them into a node where the frequency is the sum of the two and add the new node back to the queue



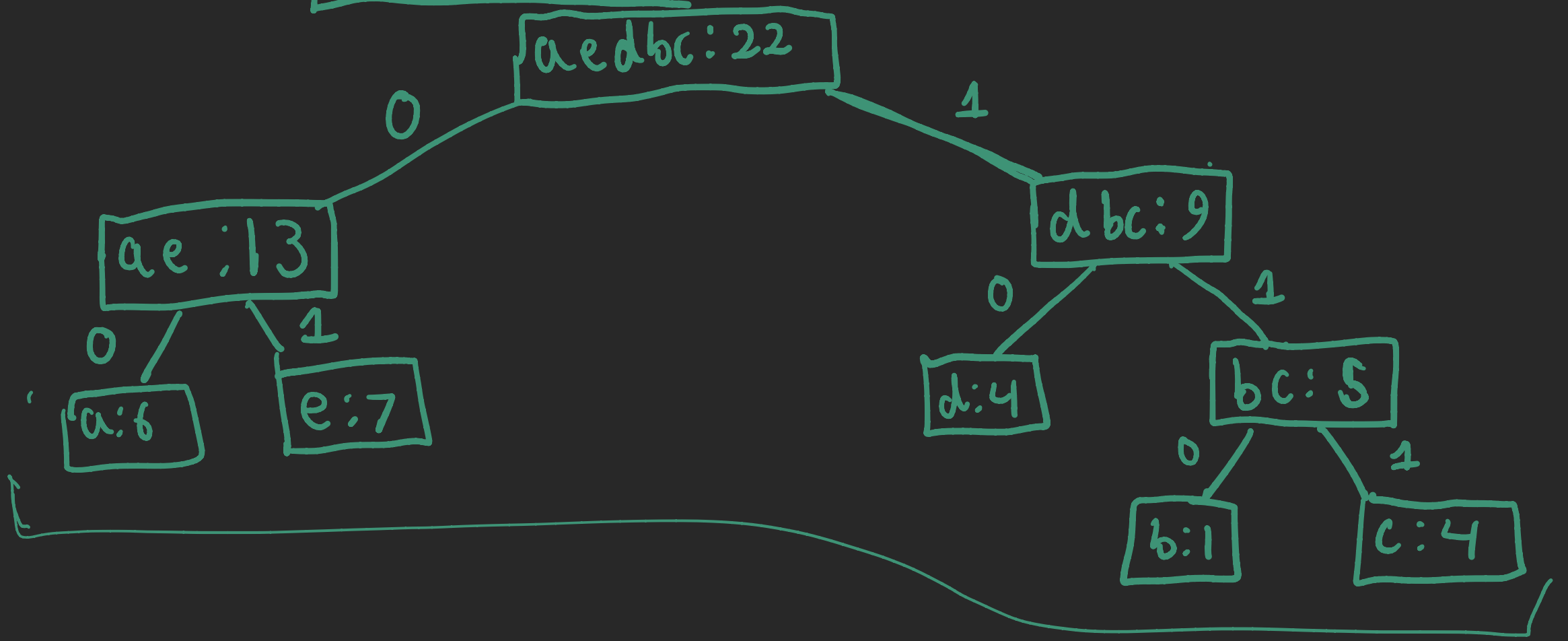
Huffman: Example

00101111
adec

debc

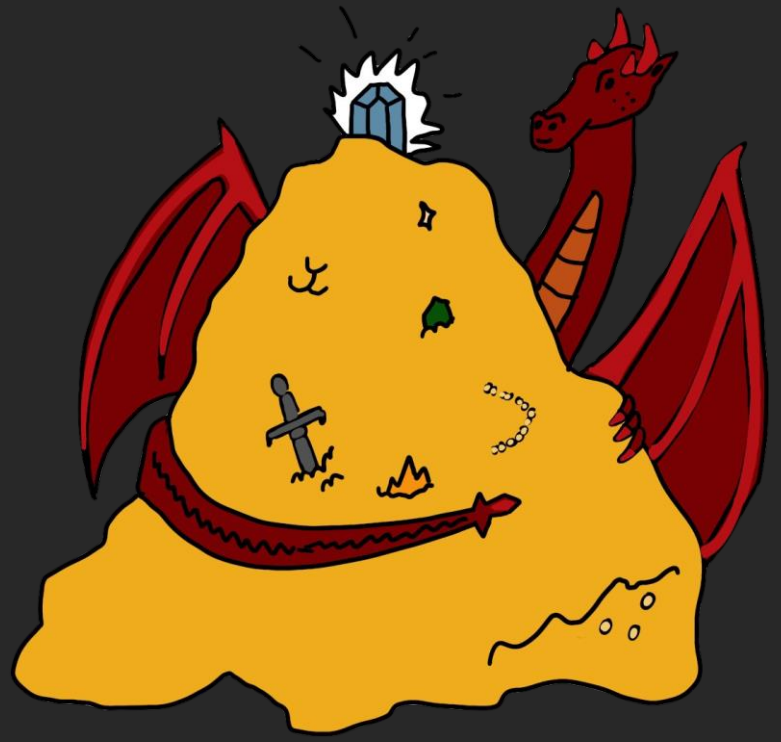
(b:1)(c:4)(d:4)(a:6)(e:7)

dbc:9



Lecture Outline

- ~~Poll review/Administrative~~
- ~~Greedy Algorithms~~
  - ~~Introduction~~
  - ~~Example~~
  - ~~Structure~~
- ~~Huffman Encoding~~
  - ~~Introduction~~
  - ~~History~~
  - ~~Terminology~~
  - ~~Example~~
- Homework examples



first call p=1, r=6

HW2B: Example 1 Quicksort

1 2 3 4 5 6

- Sort A = {5, 8, 4, 2, 7, 6} using the deterministic quicksort algorithm. Note!! Not what p, r, and q are at every call to quicksort x, i

1) p=1, r=6, x=6, q=4

2) p=1, r=3, x=2, q=1

3) p=2, r=3

4) p=5, r=6, x=8, i=4

Current array

2, 4, 5, 6, 7, 8

```
Quicksort(A,p,r) {
   if (p<r){
      q = Partition(A,p,r)
      Quicksort(A,p,q-1)
      Quicksort(A,q+1,r)
}}
```

```
Partition(A,p,r) {
   x = A[r]
   i = p-1
   for (j=p; j<=r-1;j++) {
      if A[j]<=x {
         i++
         exchange(A[i],A[j])
   }}
   exchange(A[i+1],A[r])
   return i+1
}
```

HW2A: Example 2

- Can we use Master Theorem to solve the following?
- $T(n) = 1000T(n/10) + n^3\log(n)$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$$

$$\lim_{n\to\infty} \frac{n^3\log(n)}{n^3 \cdot n^\varepsilon}$$

$$\lim_{n\to\infty} \frac{\log(n)}{n^\varepsilon}$$

$$\lim_{n\to\infty} \frac{1}{\varepsilon n^\varepsilon} = 0$$

$$\downarrow \frac{d}{dn}$$

$$\lim_{n\to\infty} \frac{\frac{1}{n}}{\varepsilon n^{\varepsilon-1}} = \frac{1}{n^1 \cdot \varepsilon n^{\varepsilon-1}} = \frac{1}{\varepsilon n^\varepsilon}$$

Master theorem. Let $a \geq 1$ and $b > 1$ be constants, let $f(n)$ be a function and let $T(n)$ be defined on the non-negative integers by the recurrence,

$$T(n) = a\,T(n/b) + f(n) \ ,$$

where we interpret n/b to mean either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$. Then $T(n)$ can be bounded asymptotically as follows:

1. If $f(n) = O(n^{\log_b a - \epsilon})$ for some constant $\epsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$
2. If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \log n)$
3. If $f(n) = \Omega(n^{\log_b a + \epsilon})$ for some constant $\epsilon > 0$, and if $a\,f(n/b) \leq c\,f(n)$ for some constant $c < 1$ and all sufficiently large n, then $T(n) = \Theta(f(n))$. □

1) $n^{\log_b a} = n^3$

2) $f(n) = n^3\log(n)$

does $f(n) = \Omega(n^{3+\varepsilon})$?