MA/CSSE 473 Day 22

Gray Code

More Decrease and Conquer Algorithms

MA/CSSE 473 Day 22

- **You can copy your answers for quiz questions 1-2 from yesterday's quiz.**
- HW 9 is due Tuesday. Its "late day" period will extend until Friday at noon.
- Don't forget the Convex Hull implementation problem. It is due 24 days after I assigned it (October 21) , and 11 of those days have already passed.
- HW 10 due Tuesday, Oct 19
- HW 11 Due Friday, Oct 22.
- Yes, there really will be 3 things due that week!
- **Student Questions**
- Gray Code
- More Decrease and Conquer algorithms

Generate all Subsets of $\{a_0, \ldots, a_{n-1}\}$

- Decrease by one (yesterday):
  - Generate S_{n-1}, the collection of the 2^{n-1} subsets of $\{a_0, \ldots, a_{n-2}\}$
  - Then $S_n = S_{n-1} \cup \{ s \cup \{a_{n-1}\} : s \in S_{n-1}\}$
- Each subset of $\{a_0, \ldots, a_{n-1}\}$ corresponds to a bit string of length n, where the i^{th} bit is 1 iff a_i is in the subset
  - Generate binary codes in numeric order (yesterday)
  - Gray code

Recap: Another approach:

- Each subset of $\{a_0, \ldots, a_{n-1}\}$ corresponds to an bit string of length n, where the i^{th} bit is 1 if and only if a_i is in the subset

```
def allSubsets(s):
  n = len(s)
  subsets=[]
  for i in range(2**n):
    subset = []
    current = i
    for j in range (n):
      if current % 2 == 1:
        subset += [s[j]]
      current /= 2
    subsets += [subset]
  return subsets
```

Output:

```
[[], [0],[1],
[0, 1], [2],
[0, 2],
[1, 2],
[0, 1, 2]]
```

Recap: Gray Code

- Named for Frank Gray
- The term can refer to any sequence of the binary codes for n bits in which only one bit changes in each transitions.
- Usually refers to binary reflected code.
- Non-binary-reflected example:
  000, 010, 011, 001, 101, 111, 110, 100
- A Gray code can be represented by its **transition sequence:** which bit changes each time
  In the above example: 1, 0, 1, 2, 1, 0, 1
- In terms of subsets, the transition sequence tells which element to add or remove from one subset to get the next subset
- **Gray code applications.** Wikipedia has a decent overview. http://en.wikipedia.org/wiki/Gray_code

Q1

Generating a Transition Sequence

- Transition sequences for Binary Reflected Code
- $T_1 = 0$
- $T_{n+1} = T_n , n, T_n^{reversed}$
- What are T_2, T_3, T_4
- T_n is always a palindrome, so $T_{n+1} = T_n , n, T_n$

Yesterday's quiz question 3 was rather silly.
The property is obvious.

Q2

Direct Recursive Gray Code Generation

- G_0 is empty
- To get G_{i+1} from G_i:
  - Start with G_i and $G_i^{reversed}$.
  - Prepend 0 to each code in the first half, and 1 to each code in the second half.
- Example: $G_2 \rightarrow G_3$
  - {00, 01, 11, 10} {10, 11, 01, 00} →
  - {000, 001, 011, 010, 110, 111, 101, 100}

Iteratively Generate Gray Code

- We add a parity bit, p.
- Set all bits (including p) to 0.

```
while True:
    printSet(a)
    p = 1 - p #flip the parity bit
    if p == 1:
        j = 0
    else:
        j = 1
        while a[j-1]==0:  # find position to the
            j += 1        # left of the rightmost 1
        if j == n:
            break
    a[j] = 1 - a[j]   # flip this bit.
```

* Based on Knuth, Volume 4, Fascicle 2, page 6.

Q3

Quote of the Day

- There are 10^11 stars in the galaxy. That used to be a huge number. But it's only a hundred billion. It's less than the national deficit!

  We used to call them astronomical numbers. Now we should call them economical numbers.
  - Richard Feynman

http://www.feynmanonline.com/

Decrease by a constant factor

Decrease by a variable amount

OTHER DECREASE-AND-CONQUER ALGORITHMS

Decrease by a Constant Factor

- **Examples that we have already seen:**
  - Binary Search
  - Exponentiation (ordinary and modular) by repeated squaring
  - Multiplication à la Russe (The Dasgupta book that I often used for the first part of the course calls it "European" instead of "Russian")
    - Example

11	13
5	26
~~2~~	~~52~~
1	104
	143

Then strike out any rows whose first number is even, and add up the remaining numbers in the second column.

Fake Coin Problem

- We have n coins
- All but one have the same weight
- One is lighter
- We have a balance scale with two pans.
- All it will tell us is whether the two sides have equal weight, or which side is heavier
- What is the minimum number of weighings that will guarantee that we find the fake coin?
- Decrease by factor of two.

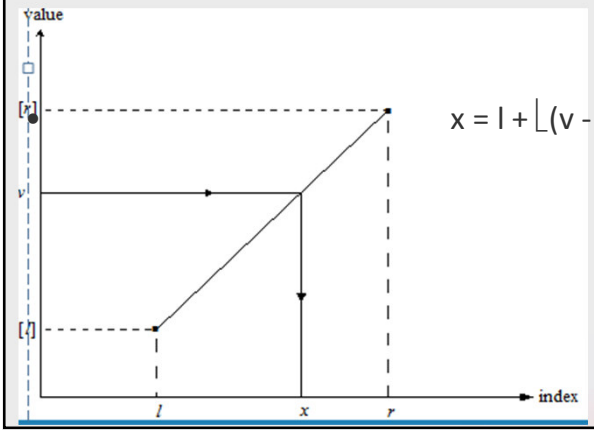
Q4

Decrease by a variable amount

- Search in a Binary Search Tree
- Interpolation Search
  - See Levitin, pp190-191
  - Also Weiss, Section 5.6.3

Interpolation Search

- Searches a sorted array similar to binary search but estimates location of the search key in A[l..r] by using its value v.
- Specifically, the values of the array's elements are assumed to grow linearly from A[l] to A[r]
- Location of v is estimated as the x-coordinate of the point on the straight line through (l, A[l]) and (r, A[r]) whose y-coordinate is v:



$$x = l + \lfloor (v - A[l])(r - l)/(A[r] - A[l]) \rfloor$$

Q5

Interpolation Search Runtime

- Average case $\Theta(\lg \lg n)$ Worst: $\Theta(n)$
- What can lead to worst-case behavior?
  - Social Security numbers of US residents
  - Phone book (Wilkes-Barre)
  - CSSE department employees, 1985-2010

adkins	criss	merkle
anderson	curry	mohan
ardis	defoe	mouck
azhar	dalkolic	mutchler
bagert	degler	oexmann
bohner	jeschke	sengupta
bowman	kaczmarczyk	shillingford
boutell	kinley	sullivan
chenoweth	laxer	surendran
chidanandan	mcleish	wollowski
clifton	mellor	young
16/33 A-D	25/33 A-D, K-M	29/33 A-D, K-M, S

Q6

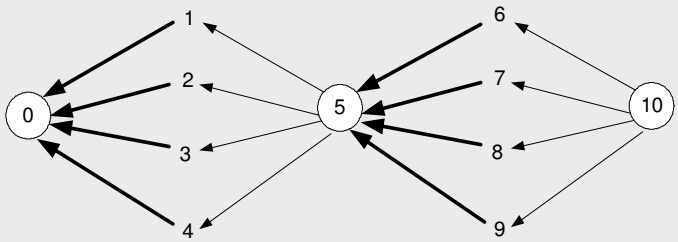
Median finding

- Find the k^{th} element of an (unordered) list of n elements
- Start with quicksort's partition method
- Informal analysis

One Pile Nim

- There is a pile of n chips. Two players take turns by removing from the pile at least 1 and at most m chips. (The number of chips taken can vary from move to move.)
- The winner is the player that takes the last chip.
- Who wins the game – the player moving first or second, if both players make the best moves possible?
- It's a good idea to analyze this and similar games "backwards", i.e., starting with $n = 0, 1, 2, \ldots$

Graph of One-Pile Nim with $m = 4$



- Vertex numbers indicate n, the number of chips in the pile. The losing position for the player to move are circled. Only winning moves from a winning position are shown (in bold).
- **Generalization:** The player moving first wins iff n is not a multiple of 5 (more generally, m+1);
  - The winning move is to take n mod 5 (n mod (m+1)) chips on every move.

Josephus problem - background

- Flavius Josephus was a Jewish general and historian who lived and wrote in the 1st century AD
- Much of what we know about 1st century life in Israel (and the beginnings of Christianity) before and after the Roman destruction of the Jewish temple in 70 AD comes from his writings
- The "Josephus problem" is based on an odd suicide pact that he describes
  - He and his men stood in a circle and counted off
  - Every other person (or every third person, accounts vary) was killed
  - The last person was supposed to kill himself
  - He must have been the next-to-last person!
  - When it got down to two people, he persuaded the other person that they should surrender instead
- http://en.wikipedia.org/wiki/Josephus

Josephus Problem

- n people, numbered 1-n, are in a circle
- Count starts with 1
- Every 2nd person is eliminated
- The last person left, J(n), is the winner
- Examples: n=8, n=7
- J(1) = 1
- Solution if n is even
- Solution if n is odd
- Use it to find J(2) … J(8)
- Clever solution: cyclic bit shift left