Lecture 10 Apr. 19. 2023

Test #1

March 01, 2023 1st hour

Asymptotic notation. $O \Omega \Theta o \omega$

Compare bounds

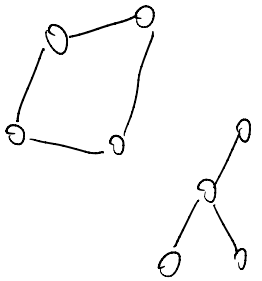
* Predict running times - tables *

* Prob Alg Impl $OP \leftarrow OI$
$X \leftarrow \Omega I$

* Comparing Problems

* Graphs special graphs cycle, tree, bi-partite, complete.

algs SLVO COLORING TSO Prim MST



Kruskal, DJSSP, EULER TOURS,
Hamiltonian paths cycles ← No Alg
MAX Ford-Fulkerson Flow.
Multi-source / sink
maximal matching

Tree Serilization

* entropy - Huffman coding

Diffie Hellman key exchanges

$\Phi(n)$

Dynamic Prog fib() LIS LCS, LED, 0-1 knapsack.

Dice Rolling

GCD. LCM $(x,y) = \frac{x \cdot y}{GCD(x,y)}$

Extended euclidean Alg $\frac{1}{n}$ % M RSA

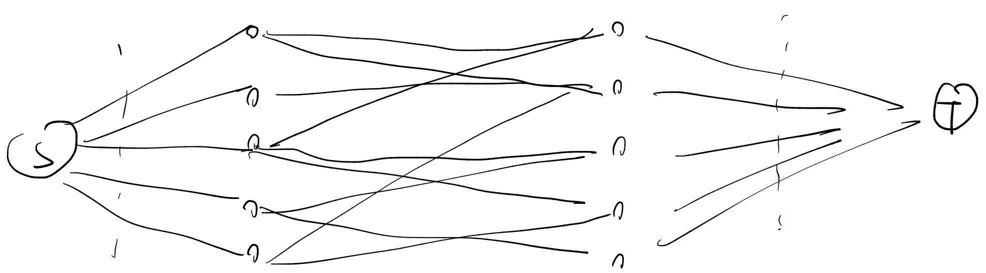
---

* Heaps * Quick Select * P & NP

* DGT. ~~LZW in jpeg~~

Maximum matching

SW HW



max flow

$\beta = 10$

386.24 $\beta = 11$

$3 \times 10^2 + 8 \times 10^1 + 6 \times 10^0 + 2 \times 10^{-1} + 4 \times 10^{-2}$

$39\overline{4}.24$

$3 \times 10^2 + 9 \times 10^1 + -4 \times 10^0 + 2 \times 10^{-1} + 4 \times 10^{-2}$

$3\overline{1}\overline{2}.1$

$3 \times 10^2 + 10 \times 10^1 + -2 \times 10^0 + 1 \times 10^{-1}$

1101 $\beta = 2$

$1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$ this is just a binary number base 2.

$= 8 + 4 + 0 + 1$

$= 13$

1101.11

$1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$

$= 8 + 4 + 0 + 1 + 0.5 + 0.25$

$= 13.75 = 13\frac{3}{4}$

251.2 $\beta = 3$ $\qquad 34\frac{2}{3}$ $\beta = 10$

$2 \times 3^2 + 5 \times 3^1 + 1 \times 3^0 + 2 \times 3^{-1}$

$= 2 \times 9 + 5 \times 3 + 1 \times 1 + \frac{2}{3}$

$= 18 + 15 + 1 + \frac{2}{3}$

$= 19 + 15 + \frac{2}{3} = 34\frac{2}{3}$

This number system here is not complete. You cannot represent negative numbers with the summer system.

27 $\quad \beta = 10 \quad D = \{0,1,2,3,4,5,6,7,8,9\}$

$3\bar{3}$ $\quad \beta = 10 \quad D = \{\bar{3}\ \bar{2}\ \bar{1}\ 0\ 1\ 2\ 3\ 4\ 5\ 6\}$

$3 \times 10^1 + -3 \times 10^2 = 30 - 3 = 27$

65 $\beta = 10 \qquad \beta = 3 \qquad D = \{\bar{1}, 0, 4\}$

$65 \% 3 = 2 \quad \bar{1}$

$\bar{1} \% 3 = 2$

$0 \% 3 = 0$

$4 \% 3 = 1$

$\bar{1}$

66

$\div 3$

$22 \% 3 = 1 \quad 4$

-4

18

$\div 3$

$6 \% 3 = 0 \quad 0$

-0

6

$\div 3$

$2 \% 3 = 2 \quad \bar{1}$

$-\bar{1}$

3

$\div 3$

$1 \% 3 = 1 \quad 4$

-4

-3

$\div 3$

$-1 \% 3 = 2 \quad \bar{1}$

$-\bar{1}$

0

$\bar{1}\ 4\ \bar{1}\ 0\ 4\ \bar{1}$

2314 β=10

2314 % 10 = 4

− 4

2310

÷ 10

231 % 10 = 1

− 1

230

÷ 10

23 % 10 = 3

− 3

20

÷ 10

2 % 10 = 2

− 2

0

2314 β=10

0 %10 0
1 1
2 2
3 3
4 4
5 5
6 6
7 7
8 8
9 9

2314

211 β=10 = β=2 D=10.14

211 % 2 = 1

− 1

210

÷ 2

105 % 2 = 1

− 1

104

÷ 2

52 % 2 = 0

− 0

52

÷ 2

26 % 2 = 0

− 0

26

÷ 2

13 % 2 = 1

− 1

12

÷ 2

6 % 2 = 0

− 0

6

÷ 2

3 % 2 = 1

− 1

2

÷ 2

1 % 2 = 1

− 1

0

1 1 0 1 0 0 1 1

① %

② −

③ ÷

⋮

② − ⇒ 0

$\beta = 10 \quad D = \{0, 1, 2 \cdots 9\}$

$-3 \quad \beta = 6$

$-3 \% 10 = 7$

-7

-10

$\div 10$

$-1 \% 10 = 9$

-9

-10

$\div 10$

$-1 \% 10 = 9$

-9

-10

$\div 10$

-1

9 9 7

-65

$-65 \% 10 = 5$

-5

-70

$\div 10$

$-7 \% 10 = 3$

-3

-10

$\div 10$

$-1 \% 10 = 9 \Rightarrow \bar{1}$

$-\bar{1}$

0

$\beta = 10 \quad D = \{0, 1, 2, \cdots 8\}$

$\bar{1}$

0

1

2

3

5

6

7

8

$\bar{1}35$

$-1 \times 10^2 + 3 \times 10^1 + 5 \times 10^0$

$= -100 + 30 + 5$

$= -65$

1 00 1 1 0 1 0 2 1 0

→ 1 1 0 1 0 1 1 0 1 0 0 1 ←

$\beta = 4$ $\{\bar{2}, \bar{1}, 0, 1, 2\}$

1 $\bar{2}$

0 2

LZW ENCODER

```
set w = NIL
loop
   read a character k
   if wk exists in the dictionary
         w = wk
   else
         output the code for w
         add wk to the dictionary
         w = k
endloop
```

A B C A B C A B C A B C

w	R	O	Add	Next w
-	A			A
A	B	65	AB	B
B	C	66	BC	C
C	A	67	CA	A
A	B			AB
AB	C	256	ABC	C
C	A			CA
CA	B	258	CAB	B
B	C			BC
BC	A	257	BCA	A
A	B			AB
AB	C			ABC
ABC	-	259	-	

0

65 = A

66 = B

67 = C

⋮

255

256 AB

257 BC

258 CA

259 ABC

260 CAB

261 BCA

96 → 63

```
read a character k
entry = dictionary entry for k
output entry
w = entry
loop
    read a character k
    entry = dictionary entry for k
    output entry
    add w + first char of entry to the dictionary
    w = entry
endloop
```

65 66 67 256 258 257 259

w	read k	entry	output	Dictionary add	Next w
-	65	A	A		A
A	66	B	B	AB	B
B	67	C	C	BC	C
C	256	AB	AB	CA	AB
AB	258	CA	CA	ABC	CA
CA	257	BC	BC	CAB	BC
BC	259	ABC	ABC	BCA	ABC
ABC					

65 = A
66 = B
67 = C
256 AB
257 BC
258 CA
259 ABC
260 CAB
261 BCA