

HW1

EE 4033 Algorithms, Fall 2018

107/10/3

Hand-Written Problems

- **Problem 1**

- Prove or disprove statements using the definition of asymptotic notations
- Do not answer only *true* or *false*
- Don't forget to find the constants n_0 , c_1 and c_2

Hand-Written Problems

- **Problem 2**

- Solve recurrences into Θ -notations
- Recurrence tree/substitution method/Master theorem

Hand-Written Problems

- **Problem 3**

- Determine if more than half of the balls inside the bag are of the same color
- If true, return the color (blue in this case)
- Solve the problem in $\Theta(n \log n)$
 - Try using methods similar to merge sort
- Solve the problem in $\Theta(n)$
 - Try pairing up the balls in groups of 2



Programming Problems

- **Problem 1**

- Sort the words in an English article by ASCII code

input.txt	output.txt
She's a 23-year-old graduate from NTU & EE118th, graduated in 2018.	23-year-old EE118th NTU She's Sheep a are either from goats graduate graduated in lambs not not pi
"Sheep" are not "goats" (not "lambs" either!)	
pi = 3.1415926	

Programming Problems

• Problem 1

- Sort the words in an English article by ASCII code

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.asciitable.com

Programming Problems

- **Problem 1**

- Sort the words in an English article by ASCII code
 - Ignore all numbers, spaces, newlines and punctuation marks
 - Compound words (e.g. *he's* or *middle-aged* or *NTUEE120th*) should be treated as a single word
 - You don't need to worry about parsing. Use the Parser class included in `tools.py`.
- Some examples
 - She's < Sheep (' = 39, e = 101)
 - NTU < NTUEE (longer words are larger)
 - 23-year-old < 23rd (- = 45, r = 114)

Programming Problems

- **Problem 1**

- You need to implement two sorting algorithms

1. Insertion Sort

- You can use the pseudocode in the textbook directly

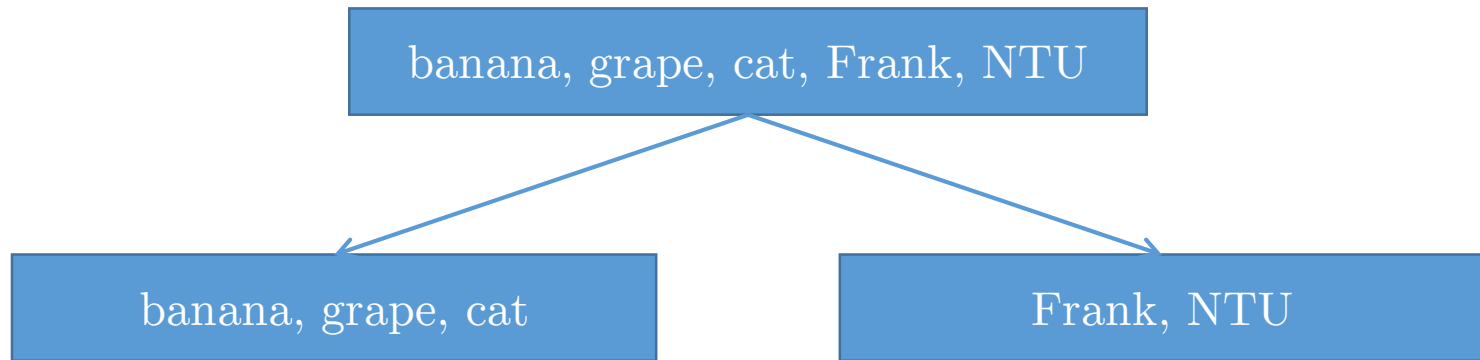
2. Merge Sort

- You need to split into **3** sub-arrays when implementing
- See the PDF for more details

Programming Problems

- **Problem 1**

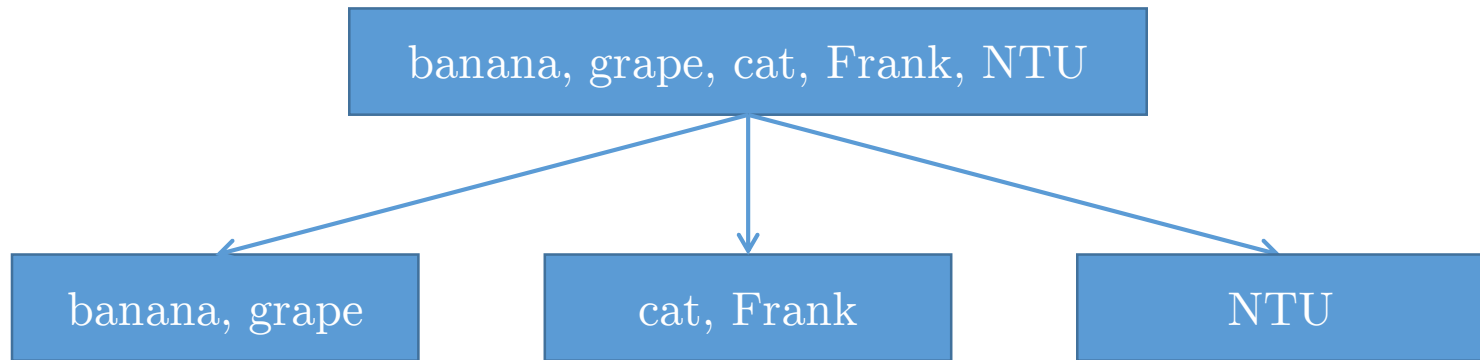
- Merge Sort: splitting into 2 sub-arrays



Programming Problems

- **Problem 1**

- Merge Sort: splitting into 3 sub-arrays

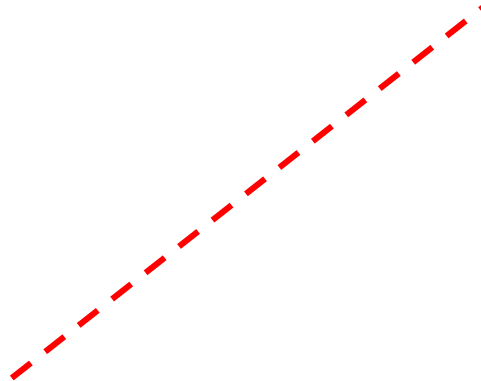


Programming Problems

- **Problem 1**

- Measure the running time of 10 input files
 - **import** time
- Plot the time-versus-size graph for both algorithms
 - Use the built-in least-squared method in Excel/MATLAB to find the fitting equation ($O(n)$? $O(n^2)$? $O(n \lg n)$?)

Running Time (t)



Input Size (n)

Programming Problems

- **Problem 2**

- Find a valley **index** i of an array
 - $A[i - 1] \geq A[i] \leq A[i + 1]$

	$A[0]$	$A[1]$	$A[2]$	$A[3]$	$A[4]$	$A[5]$	$A[6]$	$A[7]$	
∞	2	6	10	12	3	7	9	12	∞

- There will always be at least one valley in any array
- How many valleys are there in this array?
- You only need to find **any** one of them
 - 0 or 4 are both correct answers

Programming Problems

- **Problem 2**

- Implement two algorithms to solve the problem
 1. $O(n)$: just making sure that you know how to use Python
 2. $O(\lg n)$: might require some thinking (D&C)

	$A[0]$	$A[1]$	$A[2]$	$A[3]$	$A[4]$	$A[5]$	$A[6]$	$A[7]$	
∞	2	6	10	12	3	7	9	12	∞

Remarks

- Read the HW instructions very carefully
 - Failing to follow them may lead to loss of credits
- All the inputs are of arbitrary size
- You can import the following modules
 - **from tools import Parser**
 - **import sys**
 - **import time**
- Use the provided `selfCheck.py` to check the format of your `.zip/.tar` file before uploading