

43
100

EC504 A1 Advanced Data Structures, Fall 2015
Midterm Exam

Name Nicholas Masella

8

1. True False (20 Points)

Answer True or False to each of the questions below. Answer true only if it is always true as stated, with no additional assumptions. No explanation is needed.

F a. An algorithm that runs in worst-case $\Theta(n^3)$ time takes longer, on the same input, than an algorithm with worst-case $\Theta(\log n)$ time.
True

b. $\sqrt[3]{n}$ is $o(n)$. True $n > n^{1/3}$

c. $2^{\log n}$ is $\Omega(\log n)$. True $\log n < 2^{\log n} = n$

d. $2^{\log n}$ is $O(n^2)$. $n^2 > n$ True

F e. $\sum_{k=1}^n \frac{1}{2^k}$ is $\Theta(n)$. $\Rightarrow \sum_{k=1}^n \left(\frac{1}{2}\right)^k = \frac{\frac{1}{2} - \frac{1}{2}^{n+1}}{\frac{1}{2} - 1} = \frac{1}{2} - \frac{1}{2}^{n+1}$ True

f. Any comparison-based sorting algorithm will take $\Theta(n \log n)$ on a sorted array of size n .
False

T g. It is possible to find the smallest $\log(n)$ elements of an unsorted array of size n in $O(n)$ time.
False

F h. Quicksort of a sorted list of n elements has a worst-case run time of $O(n \log n)$.
True

F i. A binary search tree of height $\lg(n)$ will always have at least n elements.
True

T j. There are 14 different binary search trees that contain only the numbers 4, 5, 6, 7.
False

c. static public C(int n){

if (n == 0) return 1;

else {

int sum = C(n/2) + C(n/2); = $T(\frac{n}{2}) + T(\frac{n}{2})$

for (int ii=1; ii != n; ii++) →

for (int jj=1; jj != n; jj++)

sum += ii + jj ~~++~~;

return sum * C(n/2) * C(n/2); →

+ $2T(\frac{n}{2})$

}

$$T(n) = T(\frac{n}{2}) + T(\frac{n}{2}) + 2T(\frac{n}{2}) + \left(c \frac{n}{n!} \right) ?$$

4

$$\Rightarrow \Theta(n^2 \log n)$$

d. static public int D(int n) {

if (n == 0) return 1;

if (D(n/2) <= D(n/4)) → $T(\frac{n}{2}) + T(\frac{n}{4})$

return D(Math.sqrt(n)) + D(Math.sqrt(n)); → $T(\sqrt{n}) + T(\sqrt{n})$

else

return D(n/2); → $T(\frac{n}{2})$

}

6

$$T(n) = T(\frac{n}{2}) + T(\frac{n}{4}) + T(2\sqrt{n}) - \Theta(n^2 \log n)$$

3. More Recurrences (16 Points)

Give the tightest upper and lower bounds you can for the following recurrences.

a. $T(n) = 8T\left(\frac{n}{2}\right) + 3$

$$n^{\log_2 8} = n^{\log_2 2^3} = \frac{\log_2 2^3}{\log_2 8} =$$

$$\Theta(n^{\log_2 8})$$

Recursion tree for $T(n)$:

- Root: $T(n)$
- Level 1: $8T(n/2)$
- Level 2: $8^2T(n/4)$
- Level 3: $8^3T(n/8)$
- Level 4: $8^4T(n/16)$
- Level 5: $8^5T(n/32)$
- Level 6: $8^6T(n/64)$
- Level 7: $8^7T(n/128)$
- Level 8: $8^8T(n/256)$
- Level 9: $8^9T(n/512)$
- Level 10: $8^{10}T(n/1024)$
- Level 11: $8^{11}T(n/2048)$
- Level 12: $8^{12}T(n/4096)$
- Level 13: $8^{13}T(n/8192)$
- Level 14: $8^{14}T(n/16384)$
- Level 15: $8^{15}T(n/32768)$
- Level 16: $8^{16}T(n/65536)$
- Level 17: $8^{17}T(n/131072)$
- Level 18: $8^{18}T(n/262144)$
- Level 19: $8^{19}T(n/524288)$
- Level 20: $8^{20}T(n/1048576)$
- Level 21: $8^{21}T(n/2097152)$
- Level 22: $8^{22}T(n/4194304)$
- Level 23: $8^{23}T(n/8388608)$
- Level 24: $8^{24}T(n/16777216)$
- Level 25: $8^{25}T(n/33554432)$
- Level 26: $8^{26}T(n/67108864)$
- Level 27: $8^{27}T(n/134217728)$
- Level 28: $8^{28}T(n/268435456)$
- Level 29: $8^{29}T(n/536870912)$
- Level 30: $8^{30}T(n/1073741824)$
- Level 31: $8^{31}T(n/2147483648)$
- Level 32: $8^{32}T(n/4294967296)$
- Level 33: $8^{33}T(n/8589934592)$
- Level 34: $8^{34}T(n/17179869184)$
- Level 35: $8^{35}T(n/34359738368)$
- Level 36: $8^{36}T(n/68719476736)$
- Level 37: $8^{37}T(n/137438953472)$
- Level 38: $8^{38}T(n/274877906944)$
- Level 39: $8^{39}T(n/549755813888)$
- Level 40: $8^{40}T(n/1099511627776)$
- Level 41: $8^{41}T(n/2199023255552)$
- Level 42: $8^{42}T(n/4398046511104)$
- Level 43: $8^{43}T(n/8796093022208)$
- Level 44: $8^{44}T(n/17592186044416)$
- Level 45: $8^{45}T(n/35184372088832)$
- Level 46: $8^{46}T(n/70368744177664)$
- Level 47: $8^{47}T(n/140737488355328)$
- Level 48: $8^{48}T(n/281474976710656)$
- Level 49: $8^{49}T(n/562949953421312)$
- Level 50: $8^{50}T(n/1125899906842624)$
- Level 51: $8^{51}T(n/2251799813685248)$
- Level 52: $8^{52}T(n/4503599627370496)$
- Level 53: $8^{53}T(n/9007199254740992)$
- Level 54: $8^{54}T(n/18014398509481984)$
- Level 55: $8^{55}T(n/36028797018963968)$
- Level 56: $8^{56}T(n/72057594037927936)$
- Level 57: $8^{57}T(n/144115188075855872)$
- Level 58: $8^{58}T(n/288230376151711744)$
- Level 59: $8^{59}T(n/576460752303423488)$
- Level 60: $8^{60}T(n/1152921504606846976)$
- Level 61: $8^{61}T(n/2305843009213693952)$
- Level 62: $8^{62}T(n/4611686018427387904)$
- Level 63: $8^{63}T(n/9223372036854775808)$
- Level 64: $8^{64}T(n/18446744073709551616)$
- Level 65: $8^{65}T(n/36893488147419103232)$
- Level 66: $8^{66}T(n/73786976294838206464)$
- Level 67: $8^{67}T(n/147573952589676412928)$
- Level 68: $8^{68}T(n/295147905179352825856)$
- Level 69: $8^{69}T(n/590295810358705651712)$
- Level 70: $8^{70}T(n/1180591620717411303424)$
- Level 71: $8^{71}T(n/2361183241434822606848)$
- Level 72: $8^{72}T(n/4722366482869645213696)$
- Level 73: $8^{73}T(n/9444732965739290427392)$
- Level 74: $8^{74}T(n/18889465931478580854784)$
- Level 75: $8^{75}T(n/37778931862957161709568)$
- Level 76: $8^{76}T(n/75557863725914323419136)$
- Level 77: $8^{77}T(n/151115727451828646838272)$
- Level 78: $8^{78}T(n/302231454903657293676544)$
- Level 79: $8^{79}T(n/604462909807314587353088)$
- Level 80: $8^{80}T(n/1208925819614629174706176)$
- Level 81: $8^{81}T(n/2417851639229258349412352)$
- Level 82: $8^{82}T(n/4835703278458516698824704)$
- Level 83: $8^{83}T(n/9671406556917033397649408)$
- Level 84: $8^{84}T(n/19342813113834066795298816)$
- Level 85: $8^{85}T(n/38685626227668133590597632)$
- Level 86: $8^{86}T(n/77371252455336267181195264)$
- Level 87: $8^{87}T(n/154742504910672534362390528)$
- Level 88: $8^{88}T(n/309485009821345068724781056)$
- Level 89: $8^{89}T(n/618970019642690137449562112)$
- Level 90: $8^{90}T(n/1237940039285380274899124224)$
- Level 91: $8^{91}T(n/2475880078570760549798248448)$
- Level 92: $8^{92}T(n/4951760157141521099596496896)$
- Level 93: $8^{93}T(n/9903520314283042199192993792)$
- Level 94: $8^{94}T(n/19807040628566084398385987584)$
- Level 95: $8^{95}T(n/39614081257132168796771975168)$
- Level 96: $8^{96}T(n/79228162514264337593543950336)$
- Level 97: $8^{97}T(n/158456325028528675187087900672)$
- Level 98: $8^{98}T(n/316912650057057350374175801344)$
- Level 99: $8^{99}T(n/633825300114114700748351602688)$
- Level 100: $8^{100}T(n/1267650600228229401496703205376)$
- Level 101: $8^{101}T(n/2535301200456458802993406410752)$
- Level 102: $8^{102}T(n/5070602400912917605986812821504)$
- Level 103: $8^{103}T(n/10141204801825835211973625643008)$
- Level 104: $8^{104}T(n/20282409603651670423947251286016)$
- Level 105: $8^{105}T(n/40564819207303340847894502572032)$
- Level 106: $8^{106}T(n/81129638414606681695789005144064)$
- Level 107: $8^{107}T(n/162259276829213363391578010288128)$
- Level 108: $8^{108}T(n/324518553658426726783156020576256)$
- Level 109: $8^{109}T(n/649037107316853453566312041152512)$
- Level 110: $8^{110}T(n/1298074214633706907132624082305024)$
- Level 111: $8^{111}T(n/2596148429267413814265248164610048)$
- Level 112: $8^{112}T(n/5192296858534827628530496329220096)$
- Level 113: $8^{113}T(n/10384593717069655257060992658440192)$
- Level 114: $8^{114}T(n/20769187434139310514121985316880384)$
- Level 115: $8^{115}T(n/41538374868278621028243970633760768)$
- Level 116: $8^{116}T(n/83076749736557242056487941267521536)$
- Level 117: $8^{117}T(n/166153499473114484112975882535043072)$
- Level 118: $8^{118}T(n/332306998946228968225951765070086144)$
- Level 119: $8^{119}T(n/664613997892457936451903530140172288)$
- Level 120: $8^{120}T(n/1329227995784915872903807060280344576)$
- Level 121: $8^{121}T(n/2658455991569831745807614120560689152)$
- Level 122: $8^{122}T(n/5316911983139663491615228241121378304)$
- Level 123: $8^{123}T(n/10633823966279326983230456482242756608)$
- Level 124: $8^{124}T(n/21267647932558653966460912964485513216)$
- Level 125: $8^{125}T(n/42535295865117307932921825928971026432)$
- Level 126: $8^{126}T(n/85070591730234615865843651857942052864)$
- Level 127: $8^{127}T(n/170141183460469231731687303715884105728)$
- Level 128: $8^{128}T(n/340282366920938463463374607431768211456)$
- Level 129: $8^{129}T(n/680564733841876926926749214863536422912)$
- Level 130: $8^{130}T(n/1361129467683753853853498429727072845824)$
- Level 131: $8^{131}T(n/2722258935367507707706996859454145691648)$
- Level 132: $8^{132}T(n/5444517870735015415413993718908291383296)$
- Level 133: $8^{133}T(n/10889035741470030830827987437816582766592)$
- Level 134: $8^{134}T(n/21778071482940061661655974875633165533184)$
- Level 135: $8^{135}T(n/43556142965880123323311949751266331066368)$
- Level 136: $8^{136}T(n/87112285931760246646623899502532662132736)$
- Level 137: $8^{137}T(n/174224571863520493293247799005065324265472)$
- Level 138: $8^{138}T(n/348449143727040986586495598010130648530944)$
- Level 139: $8^{139}T(n/696898287454081973172991196020261297061888)$
- Level 140: $8^{140}T(n/1393796574908163946345982392040522594123776)$
- Level 141: $8^{141}T(n/2787593149816327892691964784081045188247552)$
- Level 142: $8^{142}T(n/5575186299632655785383929568162090376495104)$
- Level 143: $8^{143}T(n/11150372599265311570767859136324180752990208)$
- Level 144: $8^{144}T(n/22300745198530623141535718272648361505980416)$
- Level 145: $8^{145}T(n/44601490397061246283071436545296723011960832)$
- Level 146: $8^{146}T(n/89202980794122492566142873090593446023921664)$
- Level 147: $8^{147}T(n/178405961588244985132285746181186892047843328)$
- Level 148: $8^{148}T(n/356811923176489970264571492362373784095686656)$
- Level 149: $8^{149}T(n/713623846352979940529142984724747568191373312)$
- Level 150: $8^{150}T(n/1427247692705959881058285969449495136382746624)$
- Level 151: $8^{151}T(n/2854495385411919762116571938898990272765493248)$
- Level 152: $8^{152}T(n/5708990770823839524233143877797980545530986496)$
- Level 153: $8^{153}T(n/11417981541647679048466287755595961091061972992)$
- Level 154: $8^{154}T(n/22835963083295358096932575511191922182123945984)$
- Level 155: $8^{155}T(n/45671926166590716193865151022383844364247891968)$
- Level 156: $8^{156}T(n/91343852333181432387730302044767688728495783936)$
- Level 157: $8^{157}T(n/182687704666362864775460604089535377456991567872)$
- Level 158: $8^{158}T(n/365375409332725729550921208179070754913983135744)$
- Level 159: $8^{159}T(n/730750818665451459101842416358141509827966271488)$
- Level 160: $8^{160}T(n/1461501637330902918203684832716283019655932542976)$
- Level 161: $8^{161}T(n/2923003274661805836407369665432566039311865085952)$
- Level 162: $8^{162}T(n/5846006549323611672814739330865132078623730171904)$
- Level 163: $8^{163}T(n/11692013098647223345629478661730264157247460343808)$
- Level 164: $8^{164}T(n/23384026197294446691258957323460528314494920687616)$
- Level 165: $8^{165}T(n/46768052394588893382517914646921056628989841375232)$
- Level 166: $8^{166}T(n/93536104789177786765035829293842113257979682750464)$
- Level 167: $8^{167}T(n/187072209578355573530071658587684226515959365500928)$
- Level 168: $8^{168}T(n/374144419156711147060143317175368453031918731001856)$
- Level 169: $8^{169}T(n/748288838313422294120286634350736906063837462003712)$
- Level 170: $8^{170}T(n/1496577676626844588240573268701473812127674924007424)$
- Level 171: $8^{171}T(n/2993155353253689176481146537402947624255349848014848)$
- Level 172: $8^{172}T(n/5986310706507378352962293074805895248510699696029696)$
- Level 173: $8^{173}T(n/11972621413014756705924586149611790497021399392059392)$
- Level 174: $8^{174}T(n/23945242826029513411849172299223580994042798784118784)$
- Level 175: $8^{175}T(n/47890485652059026823698344598447161988085597568237568)$
- Level 176: $8^{176}T(n/95780971304118053647396689196894323976171195136475136)$
- Level 177: $8^{177}T(n/191561942608236107294793378393788647952342390272950272)$
- Level 178: $8^{178}T(n/383123885216472214589586756787577295904684780545900544)$
- Level 179: $8^{179}T(n/766247770432944429179173513575154591809369561091801088)$
- Level 180: $8^{180}T(n/1532495540865888858358347027150309183618739122183602176)$
- Level 181: $8^{181}T(n/3064991081731777716716694054300618367237478244367204352)$
- Level 182: $8^{182}T(n/6129982163463555433433388108601236734474956488734408704)$
- Level 183: $8^{183}T(n/12259964326927110866866776217202473468949912977468817408)$
- Level 184: $8^{184}T(n/24519928653854221733733552434404946937899825954937634816)$
- Level 185: $8^{185}T(n/49039857307708443467467104868809893875799651909875269632)$
- Level 186: $8^{186}T(n/98079714615416886934934209737619787751599303819750539264)$
- Level 187: $8^{187}T(n/196159429230833773869868419475239575503198607639501078528)$
- Level 188: $8^{188}T(n/392318858461667547739736838950479151006397215279002157056)$
- Level 189: $8^{189}T(n/784637716923335095479473677900958302012794430558004314112)$
- Level 190: $8^{190}T(n/1569275433846670190958947355801916604025588861116008628224)$
- Level 191: $8^{191}T(n/3138550867693340381917894711603833208051177722232017256448)$
- Level 192: $8^{192}T(n/6277101735386680763835789423207666416102355444464034512896)$
- Level 193: $8^{193}T(n/12554203470773361527671578846415332832204710888928069025792)$
- Level 194: $8^{194}T(n/25108406941546723055343157692830665664409421777856138051584)$
- Level 195: $8^{195}T(n/50216813883093446110686315385661331328818843555712276103168)$
- Level 196: $8^{196}T(n/100433627766186892221372630771322662657637687111424552206336)$
- Level 197: $8^{197}T(n/200867255532373784442745261542645325315275374222849104412672)$
- Level 198: $8^{198}T(n/401734511064747568885490523085290650630550748445698208825344)$
- Level 199: $8^{199}T(n/803469022129495137770981046170581301261101496891396417650688)$
- Level 200: $8^{200}T(n/1606938044258990275541962092341162602522202993782792835301376)$
- Level 201: $8^{201}T(n/3213876088517980551083924184682325205044405987565585670602752)$
- Level 202: $8^{202}T(n/6427752177035961102167848369364650410088811975131171341205504)$
- Level 203: $8^{203}T(n/12855504354071922204335696738729300820177623950262342682411008)$
- Level 204: $8^{204}T(n/25711008708143844408671393477458601640355247900524685364822016)$
- Level 205: $8^{205}T(n/51422017416287688817342786954917203280710495801049370729644032)$
- Level 206: $8^{206}T(n/102844034832575377634685573909834406561420991602098741459288064)$
- Level 207: $8^{207}T(n/205688069665150755269371147819668813122841983204197482918576128)$
- Level 208: $8^{208}T(n/411376139330301510538742295639337626245683966408394965837152256)$
- Level 209: $8^{209}T(n/822752278660603021077484591278675252491367932816789931674304512)$
- Level 210: $8^{210}T(n/1645504$

4. Short Answer (15 points)

A cyclic permutation of an array involves moving the first element to the end of the array. A cyclically sorted array is one that can be cyclically permuted (possibly several times) to produce a sorted array. For example, $\langle 6, 8, 10, 1, 2, 3 \rangle$ is a cyclically sorted array because it can be cyclically permuted 3 times to produce a sorted array:

$\langle 6, 8, 10, 1, 2, 3 \rangle \rightarrow \langle 8, 10, 1, 2, 3, 6 \rangle \rightarrow \langle 10, 1, 2, 3, 4, 6, 8 \rangle \rightarrow \langle 1, 2, 3, 4, 6, 8, 10 \rangle$

For each of the following, provide an answer and a short explanation:

- a. What is the fastest worst-case asymptotic run time needed to find the minimum element of an arbitrary cyclically sorted array?

- The fastest worst-case runtime would be $\Theta(n)$ since the worst case would need to go through the entire array & move the first element to the end of the array & shift the elements to the right. $\frac{n!}{(n-k)!}$

- b. What is the fastest worst-case asymptotic run time needed to find the median element of an arbitrary cyclic sorted array?

- The runtime would be $\Theta(n^{\log_2 n})$ since it would need to sort the whole array & then find the median which would be $T(\frac{n}{2}) = \Theta(n)$, $n^{\log_2 2} \leq n$

- c. What is the fastest worst-case asymptotic run time needed to sort the elements of an arbitrary cyclically sorted array?

- The fastest worst case runtime would be $\Theta(1)$ since it would be the case where the array is already sorted.

11/25

5. Short answer (25 points)

For each of the following, provide an answer and a short explanation:

- a. Determine the expected time to insert a uniformly random integer between 0 and n into a max heap containing the numbers 0 through $n/2$.

- The runtime would be $\Theta(\log(n))$ since would need to only bubble up and check half the tree if necessary

-5

- b. Determine the worst case time required to sort a list of $\log n$ distinct elements in the range $\{1, \dots, n^2\}$ with counting sort.

- It would be $\Theta(\log n \cdot n^2)$ which would give a worst case runtime of $\Theta(n^2)$ since it would need to go through array & have an unique index of $\log n$ element & then go through the range $1 \rightarrow n^2$

- c. Determine the worst case time required to sort a list of $\log n$ distinct elements in the range $\{1, \dots, n^2\}$ with insertion sort.

- The worst case would be $\Theta((\log n)^2)$

- d. Determine the best case time required to insert n^2 elements into an empty max heap.

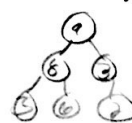
- The best case would be $\Theta(1)$ since it is a lazy data structure and will just add all elements to heap w/o sorting

-4

2 3 5 6 8 9

- e. If you insert the numbers 9, 8, 3, 5, 6, 2 into an initially empty min heap, what will the heap look like?

- Initially it will violate the min heap property but would need to bubble & sort heap ok



-5

- Mid term Rev -

- Part #1 T/F

d) $2 \lg n$ is $\Theta(n^2)$ \Rightarrow True $\log_{10} 2 \Rightarrow$ True

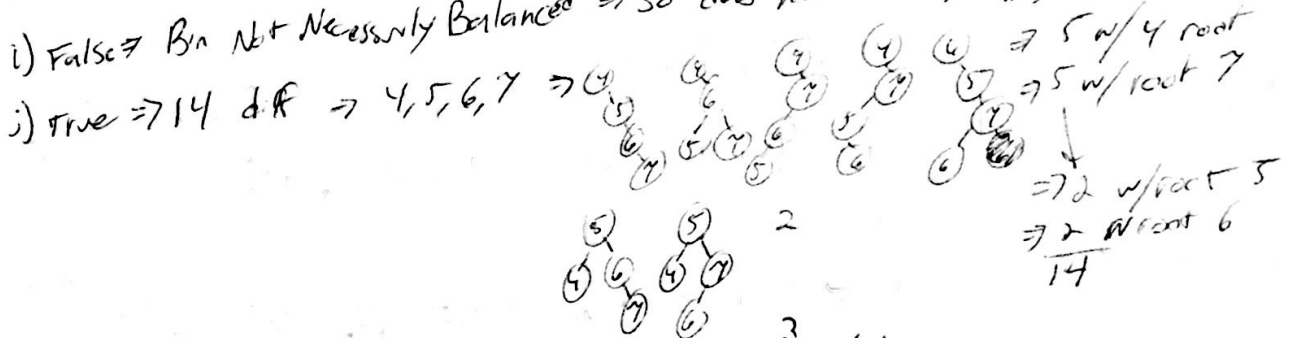
e) $\sum_{k=1}^n \frac{1}{2^k}$ is $\Theta(n)$ $\Rightarrow \sum_{k=1}^n \frac{1}{2^k} = \sum_{k=1}^n \left(\frac{1}{2}\right)^k \leq \sum_{k=0}^{\infty} \left(\frac{1}{2}\right)^k = \frac{1}{1-\frac{1}{2}} = 2$

f) False \Rightarrow Bubble sort $\Rightarrow \Theta(n)$ = not always true

g) True \Rightarrow Linear Time Select algo \Rightarrow can select it very fast

h) False $\Rightarrow \Theta(n^2)$

i) False \Rightarrow Bin Not Necessarily Balanced \Rightarrow so does not always apply



(2) a) $T(n) = 2T\left(\frac{n}{2}\right) + \Theta(1) \Rightarrow$ Master Case #1 $\Rightarrow \Theta(1) \Theta(n)$

b) $T(n) = 4T\left(\frac{n}{4}\right) + \Theta(1) \Rightarrow$ Annihilates $\Rightarrow \Theta(4^n)$

c) $T(n) = 4T\left(\frac{n}{2}\right) + n^2 \Rightarrow$ Master Case #2 $\Rightarrow \Theta(n^{\log_2 4} \log n)$, $\Theta(n^2 \log n)$

d) Hard to find bounds gave 6 points \Rightarrow Prof could find bounds

(3) a) $T(n) = 8T\left(\frac{n}{2}\right) + 3 \Rightarrow$ Master Case #1 $\Rightarrow \Theta(n^3)$

b) $T(n) = T(n-1) + n \Rightarrow$ Iteration or $\Rightarrow T(n-2) + n-1 + n, \dots \Rightarrow T(1) + \sum_{i=1}^n i \Rightarrow \Theta(n^2)$
 (not sure, ignore)
 (harder)

c) Master Case #1 $\Rightarrow \Theta(n^{\log_2 2}) \Rightarrow \Theta(n)$

d) Take log of both sides, get: $\log_2 T(n) = 2 \log_2 T(n-1) - \log_2 T(n-2)$

\Rightarrow substitute $s(n) = \log_2 T(n)$
 $s(n) = 2s(n-1) - s(n-2) \Rightarrow$ Annihilates $\Rightarrow s(n) = (\alpha n + \beta) \cdot 1^n$

$\Rightarrow T(n) = 2^{\alpha n} \cdot 2^{\beta} \Rightarrow$ use initial conditions: $\beta = 0, \alpha = \log_2 \pi$
 $\Rightarrow \Theta(\pi^n)$

4) Challenging Question

⇒ Can move in both or one direction, doesn't really matter

a) $\Theta(\log n)$ ⇒ could do a modified Binary search ⇒ looking for
breaking point of 2 sorted arrays
ie) 3rd array $\langle 10, 1, 2, 3, 6, 8 \rangle$

b) $\Theta(\log n)$ ⇒ if find median know how many elements in array

c) $\Theta(n)$ ⇒ don't have to run sorty algo since already sorted

5) a) Heap has $\Theta(\frac{n}{2})$ elements; having a 50% chance of going to ~~the~~ insert
as root, having 50% that it could go into diff levels of heap

b) Can't sort routine $\Theta(n+m)$ ⇒ $\Theta(\log n + n^2)$ ⇒ $\Theta(n^2)$
↑ ↑
Heap n^2

c) Insertion $\Theta(n^2)$ worst ⇒ if have $\log n$ elements then $\Theta((\log n)^2)$

d) Best case to insert one element ⇒ $\Theta(1)$ ⇒ n^2 times ⇒ $\Theta(n^2)$

e) Min Heap ⇒