[算法分析与设计](javascript:;)

窗体顶端

窗体底端

* [试卷自测](javascript:void(0))

**一、单选题 （题数：136，考试抽取20道，分值20分）**

1

折半查找长度为n的线性表,平均查找长度为()

窗体顶端

* A、[n](javascript:void(0);)
* B、[log n](javascript:void(0);)
* C、[nlogn](javascript:void(0);)
* D、[(n+1)/2](javascript:void(0);)

窗体底端

正确答案： B

2

中印战争西山口战役刘伯承提出的“打头、截尾、剖腹、击背” 是()思想。

窗体顶端

* A、[贪心](javascript:void(0);)
* B、[分治](javascript:void(0);)
* C、[递推](javascript:void(0);)
* D、[枚举](javascript:void(0);)

窗体底端

正确答案： B

3

确定第 i 阶段的收益函数和从第 i 阶段出发到第n 阶段所获得收益的最优值,建立动态规划基本方程。这种方法是()

窗体顶端

* A、[正推](javascript:void(0);)
* B、[反推](javascript:void(0);)
* C、[自顶向下](javascript:void(0);)
* D、[自底向上](javascript:void(0);)

窗体底端

正确答案： B

4

通常我们讲的时间复杂度是( )情况下的时间复杂度。

窗体顶端

* A、[平均](javascript:void(0);)
* B、[最好](javascript:void(0);)
* C、[最坏](javascript:void(0);)
* D、[任意](javascript:void(0);)

窗体底端

正确答案： C

5

QQ截图20190624101057.png

窗体顶端

* A、[θ](javascript:void(0);)
* B、[O](javascript:void(0);)
* C、[[QQ截图20190624094306.png](javascript:void(0);)](javascript:void(0);)
* D、[o](javascript:void(0);)

窗体底端

正确答案： D

6

0-1背包问题的枚举算法，如果在百万次每秒的计算机上运行，1年可以计算的问题规模估计是？

窗体顶端

* A、[40](javascript:void(0);)
* B、[60](javascript:void(0);)
* C、[30](javascript:void(0);)
* D、[50](javascript:void(0);)

窗体底端

正确答案： A

7

下面不是影响回溯算法效率的主要因素的是()

窗体顶端

* A、[满足约束函数的x[k]值的个数](javascript:void(0);)
* B、[x[k]的搜索顺序](javascript:void(0);)
* C、[优先级](javascript:void(0);)
* D、[上界函数约束的所有x[k]的个数](javascript:void(0);)

窗体底端

正确答案： C

8

使用分治法求解不需要满足的条件是（ ）。

窗体顶端

* A、[子问题不能够重复](javascript:void(0);)
* B、[子问题的解可以合并](javascript:void(0);)
* C、[原问题和子问题使用相同的方法求解](javascript:void(0);)
* D、[子问题必须是一样的](javascript:void(0);)

窗体底端

正确答案： D

9

以下不可以使用分治法求解的是( )。

窗体顶端

* A、[循环赛日程表](javascript:void(0);)
* B、[最接近点对问题](javascript:void(0);)
* C、[最大子段和问题](javascript:void(0);)
* D、[0/1背包问题](javascript:void(0);)

窗体底端

正确答案： D

10

Floyd算法的时间复杂度为O()

窗体顶端

* A、[mn](javascript:void(0);)
* B、[m+nlgn](javascript:void(0);)
* C、[mlogn](javascript:void(0);)
* D、[n^3](javascript:void(0);)

窗体底端

正确答案： D

11

堆排序的时间复杂度是O（）。

窗体顶端

* A、[O（n）](javascript:void(0);)
* B、[2](javascript:void(0);)[n](javascript:void(0);)
* C、[n](javascript:void(0);)[2](javascript:void(0);)
* D、[nlogn](javascript:void(0);)

窗体底端

正确答案： D

12

随机快速排序的时间复杂度是()。

窗体顶端

* A、[O(n)](javascript:void(0);)
* B、[[QQ截图20190624103116.png](javascript:void(0);)](javascript:void(0);)
* C、[[QQ截图20190624103131.png](javascript:void(0);)](javascript:void(0);)
* D、[O(nlogn)](javascript:void(0);)

窗体底端

正确答案： D

13

下面关于贪心算法错误的是()

窗体顶端

* A、[贪心算法总能找到可行解,并且是最优解。](javascript:void(0);)
* B、[问题的最优子结构性质是该问题可用贪心算法或动态规划算法求解的关键特征。](javascript:void(0);)
* C、[贪心算法一般预处理后再进行最优化选择。](javascript:void(0);)
* D、[贪心选择中每一步的局部最优解都构成全局最优解的一部分.](javascript:void(0);)

窗体底端

正确答案： A

14

下面有关递归与循环的说法错误的是()

窗体顶端

* A、[递归是比循环更灵活的重复操作的机制。](javascript:void(0);)
* B、[递归是一种比循环更强、更好用的实现“重复操作”的机制。](javascript:void(0);)
* C、[当问题需要“后进先出”的操作时,用递归算法更有效。](javascript:void(0);)
* D、[递归方法相比循环方法大大地减少了算法的计算量。](javascript:void(0);)

窗体底端

正确答案： D

15

下面不是备忘录算法特点的是()

窗体顶端

* A、[递推关系](javascript:void(0);)
* B、[自顶向下计算](javascript:void(0);)
* C、[最优子结构](javascript:void(0);)
* D、[子问题独立](javascript:void(0);)

窗体底端

正确答案： D

16

含负权的最短路问题一般使用（）求解。

窗体顶端

* A、[动态规划](javascript:void(0);)
* B、[贪心算法](javascript:void(0);)
* C、[分治算法](javascript:void(0);)
* D、[网络流算法](javascript:void(0);)

窗体底端

正确答案： A

17

未来与过去无关,指的是()性质。

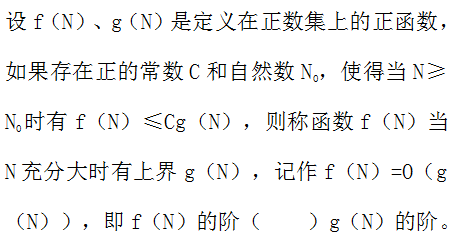
窗体顶端

* A、[贪心选择](javascript:void(0);)
* B、[无后效性](javascript:void(0);)
* C、[最优子结构](javascript:void(0);)
* D、[重叠子问题](javascript:void(0);)

窗体底端

正确答案： B

18



窗体顶端

* A、[不高于](javascript:void(0);)
* B、[不低于](javascript:void(0);)
* C、[等价于](javascript:void(0);)
* D、[逼近](javascript:void(0);)

窗体底端

正确答案： A

19

减少子问题合并的时间,就是减少时间复杂度函数T(n)=aT(n/b)+f(n) 中的()值。

窗体顶端

* A、[a](javascript:void(0);)
* B、[b](javascript:void(0);)
* C、[f(n)](javascript:void(0);)
* D、[n](javascript:void(0);)

窗体底端

正确答案： C

20

动态规划方法使用（ ）计算方式。

窗体顶端

* A、[自顶向下](javascript:void(0);)
* B、[自高到低](javascript:void(0);)
* C、[自低到高](javascript:void(0);)
* D、[自底向上](javascript:void(0);)

窗体底端

正确答案： D

21

下列算法中，通常以深度优先方式系统搜索问题解的是（     ）。

窗体顶端

* A、[备忘录法](javascript:void(0);)
* B、[动态规划法](javascript:void(0);)
* C、[贪心法](javascript:void(0);)
* D、[回溯法](javascript:void(0);)

窗体底端

正确答案： D

22

把任意一个最优解逐渐变为贪心算法的解,不会影响其最优性。这种证明方法是()

窗体顶端

* A、[领先](javascript:void(0);)
* B、[反证](javascript:void(0);)
* C、[交换论证](javascript:void(0);)
* D、[界](javascript:void(0);)

窗体底端

正确答案： C

23

快速排序的时间复杂度是O（）。

窗体顶端

* A、[n](javascript:void(0);)
* B、[2](javascript:void(0);)[n](javascript:void(0);)
* C、[n](javascript:void(0);)[2](javascript:void(0);)
* D、[nlogn](javascript:void(0);)

窗体底端

正确答案： C

24

Dinic算法的时间复杂度为（）

窗体顶端

* A、[mn](javascript:void(0);)[2](javascript:void(0);)
* B、[mn](javascript:void(0);)
* C、[m](javascript:void(0);)[2](javascript:void(0);)[n](javascript:void(0);)
* D、[m](javascript:void(0);)[2](javascript:void(0);)[logC](javascript:void(0);)

窗体底端

正确答案： A

25

便于实现集合操作的子集生成算法是（  ）

窗体顶端

* A、[位向量法](javascript:void(0);)
* B、[二进制法](javascript:void(0);)
* C、[增量构造法](javascript:void(0);)

窗体底端

正确答案： B

26

设有5000个无序的元素，希望用最快的速度挑选出其中前10个最大的元素，最好选用(   )法。

窗体顶端

* A、[冒泡排序](javascript:void(0);)
* B、[快速排序](javascript:void(0);)
* C、[合并排序](javascript:void(0);)
* D、[基数排序](javascript:void(0);)

窗体底端

正确答案： A

27

从所有候选答案中去搜索正确的解，这是 （）算法。

窗体顶端

* A、[蛮力](javascript:void(0);)
* B、[枚举](javascript:void(0);)
* C、[递推](javascript:void(0);)

窗体底端

正确答案： B

28

在下列算法中有时找不到问题解的是（    ）。

窗体顶端

* A、[蒙特卡罗算法](javascript:void(0);)
* B、[拉斯维加斯算法](javascript:void(0);)
* C、[舍伍德算法](javascript:void(0);)
* D、[数值随机算法](javascript:void(0);)

窗体底端

正确答案： B

29

从大规模问题逐步化为小规模问题的算法是（）

窗体顶端

* A、[递归](javascript:void(0);)
* B、[正推](javascript:void(0);)
* C、[倒推](javascript:void(0);)
* D、[迭代](javascript:void(0);)

窗体底端

正确答案： A

30

下面不是证明贪心算法证明方法的有（）。

窗体顶端

* A、[领先](javascript:void(0);)
* B、[优化](javascript:void(0);)
* C、[交换论证](javascript:void(0);)
* D、[界](javascript:void(0);)

窗体底端

正确答案： B

31

P类问题可以（ ）

窗体顶端

* A、[多项式时间计算](javascript:void(0);)
* B、[指数时间计算](javascript:void(0);)
* C、[指数时间验证](javascript:void(0);)

窗体底端

正确答案： A

32

下面描述错误的是()

窗体顶端

* A、[有多项式时间算法的问题是易解问题](javascript:void(0);)
* B、[[QQ截图20190624102907.png](javascript:void(0);)](javascript:void(0);)
* C、[[QQ截图20190624102950.png](javascript:void(0);)](javascript:void(0);)
* D、[[QQ截图20190624103007.png](javascript:void(0);)](javascript:void(0);)

窗体底端

正确答案： C

33

下面关于算法的说法错误的是()。

窗体顶端

* A、[同一数学模型使用不同的数据结构会有不同的算法,有效性有很大差别。](javascript:void(0);)
* B、[证明算法不正确,只需给出一个反例,算法不能正确处理即可。](javascript:void(0);)
* C、[算法是一个语句集合,按照顺序执行语句,处理实例,得到正确答案。](javascript:void(0);)
* D、[同一算法只有一种形式描述。](javascript:void(0);)

窗体底端

正确答案： D

34

最大独立集问题,如果在10亿次每秒的计算机上运行,当n=50时,需要计算的时间估计是?

窗体顶端

* A、[1小时](javascript:void(0);)
* B、[24小时](javascript:void(0);)
* C、[1年](javascript:void(0);)
* D、[100年](javascript:void(0);)

窗体底端

正确答案： D

35

http://p.ananas.chaoxing.com/star3/origin/b24e5d54ce70e5094f497f44cd3679a1.png则f(n)=      (g(n))

窗体顶端

* A、[Ο](javascript:void(0);)
* B、[Ω](javascript:void(0);)
* C、[ω](javascript:void(0);)
* D、[o](javascript:void(0);)

窗体底端

正确答案： C

36

分块查找256个元素的数组,每块的最佳长度是\_\_\_。

窗体顶端

* A、[8](javascript:void(0);)
* B、[16](javascript:void(0);)
* C、[32](javascript:void(0);)
* D、[64](javascript:void(0);)

窗体底端

正确答案： C

37此题重复

Floyd算法的时间复杂度为O()

窗体顶端

* A、[mn](javascript:void(0);)
* B、[m+nlgn](javascript:void(0);)
* C、[mlogn](javascript:void(0);)
* D、[[QQ截图20190624092906.png](javascript:void(0);)](javascript:void(0);)

窗体底端

正确答案： D

38

0-1背包问题的枚举算法,如果在万亿次每秒的计算机上运行,1年可以计算的问题规模估计是?

窗体顶端

* A、[40](javascript:void(0);)
* B、[60](javascript:void(0);)
* C、[30](javascript:void(0);)
* D、[50](javascript:void(0);)

窗体底端

正确答案： B

39

算法复杂度分析的两种基本方法为（  ）和（    ）。

窗体顶端

* A、[结构化方法 面向对象方法](javascript:void(0);)
* B、[事后统计  事前分析](javascript:void(0);)
* C、[几何复杂度  平均复杂度](javascript:void(0);)
* D、[平摊复杂度 平滑复杂度](javascript:void(0);)

窗体底端

正确答案： B

40

下面哪种函数不是回溯法中为避免无效搜索采取的策略

窗体顶端

* A、[递归函数](javascript:void(0);)
* B、[剪枝函数](javascript:void(0);)
* C、[限界函数](javascript:void(0);)
* D、[约束函数](javascript:void(0);)

窗体底端

正确答案： A

41

使目标函数最大（小）的解是问题的（）

窗体顶端

* A、[最优解](javascript:void(0);)
* B、[可行解](javascript:void(0);)

窗体底端

正确答案： A

42

log10=( )(10)

窗体顶端

* A、[θ](javascript:void(0);)
* B、[O](javascript:void(0);)
* C、[W](javascript:void(0);)
* D、[o](javascript:void(0);)

窗体底端

正确答案： A

43

对于含有n个元素的子集树问题,最坏情况下其解空间的叶结点数目为( )。

窗体顶端

* A、[n!](javascript:void(0);)
* B、[2^n](javascript:void(0);)
* C、[2^(n+1)-1](javascript:void(0);)
* D、[2^(n-1)](javascript:void(0);)

窗体底端

正确答案： B

44有点重复

下面说法关于算法与问题的说法错误的是（）。

窗体顶端

* A、[如果一个算法能应用于问题的任意实例，并保证得到正确解答，称这个算法解答了该问题。](javascript:void(0);)
* B、[算法是一种计算方法，对问题的每个实例计算都能得到正确答案。](javascript:void(0);)
* C、[同一问题可能有几种不同的算法，解题思路和解题速度也会显著不同。](javascript:void(0);)
* D、[证明算法不正确，需要证明对任意实例算法都不能正确处理。](javascript:void(0);)

窗体底端

正确答案： D

45

QQ截图20190624104218.png

窗体顶端

* A、[θ](javascript:void(0);)
* B、[O](javascript:void(0);)
* C、[[QQ截图20190624094306.png](javascript:void(0);)](javascript:void(0);)
* D、[o](javascript:void(0);)

窗体底端

正确答案： D

46重复

快速排序的时间复杂度是O（）。

窗体顶端

* A、[n](javascript:void(0);)
* B、[2](javascript:void(0);)[n](javascript:void(0);)
* C、[n](javascript:void(0);)[2](javascript:void(0);)
* D、[nlogn](javascript:void(0);)

窗体底端

正确答案： C

47

回溯算法和分支限界法的问题的解空间树不会是( )。

窗体顶端

* A、[有序树](javascript:void(0);)
* B、[子集树](javascript:void(0);)
* C、[排列树](javascript:void(0);)
* D、[无序树](javascript:void(0);)

窗体底端

正确答案： D

48重复

对于含有n个元素的子集树问题,最坏情况下其解空间的叶结点数目为( )。

窗体顶端

* A、[n!](javascript:void(0);)
* B、[[QQ截图20190624092713.png](javascript:void(0);)](javascript:void(0);)
* C、[[QQ截图20190624092802.png](javascript:void(0);)](javascript:void(0);)
* D、[2^(n-1)](javascript:void(0);)

窗体底端

正确答案： B

49

下面有关递归与迭代的说法错误的是（）

窗体顶端

* A、[递归与迭代都是解决“重复操作”的机制。](javascript:void(0);)
* B、[递归算法的实现往往要比迭代算法耗费更多的时间。](javascript:void(0);)
* C、[每个迭代算法原则上总可以转换成与它等价的递归算法。](javascript:void(0);)
* D、[每个递归算法原则上总可以转换成与它等价的迭代算法](javascript:void(0);)

窗体底端

正确答案： D

50

在一般输入数据的程序里，输入多少会影响到算法的计算复杂度，为了消除这种影响可用（      ）对输入进行预处理。

窗体顶端

* A、[蒙特卡罗算法](javascript:void(0);)
* B、[拉斯维加斯算法](javascript:void(0);)
* C、[舍伍德算法](javascript:void(0);)
* D、[数值随机化算法](javascript:void(0);)

窗体底端

正确答案： C

51

动态规划和分治算法的共同点是()

窗体顶端

* A、[最优子结构性质](javascript:void(0);)
* B、[重叠子问题性质](javascript:void(0);)
* C、[加速原理](javascript:void(0);)
* D、[贪婪准则](javascript:void(0);)

窗体底端

正确答案： A

52

贪心算法基本要素有（   ）和最优子结构性质。

窗体顶端

* A、[分解合并性质](javascript:void(0);)
* B、[独立子问题性质](javascript:void(0);)
* C、[贪心选择性质](javascript:void(0);)
* D、[叠子问题性质](javascript:void(0);)

窗体底端

正确答案： C

53

下面哪种函数是回溯法中为避免无效搜索采取的策略（        ）

窗体顶端

* A、[递归函数](javascript:void(0);)
* B、[剪枝函数](javascript:void(0);)
* C、[随机数函数](javascript:void(0);)
* D、[搜索函数](javascript:void(0);)

窗体底端

正确答案： B

54

在对问题的解空间树进行搜索的方法中，一个活结点最多有一次机会成为活结点的是（    ）。

窗体顶端

* A、[回溯法](javascript:void(0);)
* B、[分支限界法](javascript:void(0);)
* C、[回溯法和分支限界法](javascript:void(0);)
* D、[回溯法求解子集树问题](javascript:void(0);)

窗体底端

正确答案： B

55

T(n) = 2T(n/2) + n2 ，T(1)=1，则 T(n) =（）

窗体顶端

* A、[Ω(n](javascript:void(0);)[3](javascript:void(0);)[)](javascript:void(0);)
* B、[O(nlogn)](javascript:void(0);)
* C、[O(n)](javascript:void(0);)
* D、[O(n](javascript:void(0);)[2](javascript:void(0);)[)](javascript:void(0);)

窗体底端

正确答案： D

56

下面有关说法错误的是()

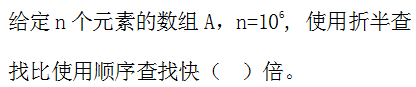
窗体顶端

* A、[倒推法是从后向前推解问题的方法.](javascript:void(0);)
* B、[有些问题采用倒推法,容易理解和解决。](javascript:void(0);)
* C、[循环用于重复性的工作。循环体的特点是:“以不变应万变”](javascript:void(0);)
* D、[高阶递推方程需要使用换元迭代化简为一阶方程求解。](javascript:void(0);)

窗体底端

正确答案： D

57



窗体顶端

* A、[5000](javascript:void(0);)
* B、[50000](javascript:void(0);)
* C、[500](javascript:void(0);)
* D、[10000](javascript:void(0);)

窗体底端

正确答案： B

58

下列算法中通常以自底向上的方式求解最优解的是()。

窗体顶端

* A、[分治法](javascript:void(0);)
* B、[动态规划法](javascript:void(0);)
* C、[贪心法](javascript:void(0);)
* D、[回溯](javascript:void(0);)

窗体底端

正确答案： B

59

下面关于NP问题说法正确的是（ ）

窗体顶端

* A、[NP问题都是不可能解决的问题](javascript:void(0);)
* B、[P类问题包含在NP类问题中](javascript:void(0);)
* C、[NP完全问题是P类问题的子集](javascript:void(0);)
* D、[NP类问题包含在P类问题中](javascript:void(0);)

窗体底端

正确答案： B

60

如果每条边的最大容量为1，则时间复杂度是O（nm）的网络流算法有（）

窗体顶端

* A、[FF算法](javascript:void(0);)
* B、[容量缩放算法](javascript:void(0);)
* C、[EK算法](javascript:void(0);)
* D、[Dinic算法](javascript:void(0);)

窗体底端

正确答案： A

61

下面关于程序和算法的说法错误的是()

窗体顶端

* A、[算法必须在有穷时间终止](javascript:void(0);)
* B、[程序是算法用某种程序设计语言的具体实现](javascript:void(0);)
* C、[程序总是在有穷步的运算后终止](javascript:void(0);)
* D、[算法可以使用自然语言描述,但要保证无歧义](javascript:void(0);)

窗体底端

正确答案： C

62重复

把任意一个最优解逐渐变为贪心算法的解,不会影响其最优性。这种证明方法是()

窗体顶端

* A、[领先](javascript:void(0);)
* B、[反证](javascript:void(0);)
* C、[交换论证](javascript:void(0);)
* D、[界](javascript:void(0);)

窗体底端

正确答案： C

63

未来与过去无关指的是（  ）的性质

窗体顶端

* A、[贪心选择](javascript:void(0);)
* B、[无后效性](javascript:void(0);)
* C、[最优子结构](javascript:void(0);)
* D、[重叠子问题](javascript:void(0);)

窗体底端

正确答案： B

64

最大独立集问题,如果在10亿次每秒的计算机上运行,100年可以计算的图的规模估计是?

窗体顶端

* A、[10](javascript:void(0);)
* B、[50](javascript:void(0);)
* C、[100](javascript:void(0);)
* D、[150](javascript:void(0);)

窗体底端

正确答案： B

65

下列说法错误的是

窗体顶端

* A、

[20190603180500.png](javascript:void(0);)

* B、

[20190603180520.png](javascript:void(0);)

* C、

[判定问题可多项式时间变换到优化问题](javascript:void(0);)

* D、

[如果一个NP完全问题有多项式时间算法，那么NP中的每一个问题都可以有多项式时间算法](javascript:void(0);)

窗体底端

正确答案： C

66

以下关于判定问题难易处理的叙述中正确的是( )。

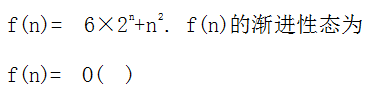
窗体顶端

* A、[可以由多项式时间算法求解的问题是难处理的。](javascript:void(0);)
* B、[需要超过多项式时间算法求解的问题是易处理的](javascript:void(0);)
* C、[可以由多项式时间算法求解的问题是易处理的](javascript:void(0);)
* D、[需要超过多项式时间算法求解的问题是不能处理的](javascript:void(0);)

窗体底端

正确答案： C

67



窗体顶端

* A、[n](javascript:void(0);)
* B、[[QQ截图20190624102629.png](javascript:void(0);)](javascript:void(0);)
* C、[[QQ截图20190624102647.png](javascript:void(0);)](javascript:void(0);)
* D、[nlogn](javascript:void(0);)

窗体底端

正确答案： C

68

下面有关枚举算法说法错误的是()

窗体顶端

* A、[枚举法适用于问题的小规模实例](javascript:void(0);)
* B、[减少枚举变量可以减少枚举算法的时间复杂度。](javascript:void(0);)
* C、[在某些问题实例中枚举是唯一的解决方法。](javascript:void(0);)
* D、[蛮力是枚举算法的一种](javascript:void(0);)

窗体底端

正确答案： D

69

QQ截图20190624094232.png

窗体顶端

* A、[θ](javascript:void(0);)
* B、[O](javascript:void(0);)
* C、[[QQ截图20190624094306.png](javascript:void(0);)](javascript:void(0);)
* D、[o](javascript:void(0);)

窗体底端

正确答案： C

70

20190603170841.png

窗体顶端

* A、[θ](javascript:void(0);)
* B、[O](javascript:void(0);)
* C、[W](javascript:void(0);)
* D、[o](javascript:void(0);)

窗体底端

正确答案： A

71

备忘录与递归算法的不同点是()

窗体顶端

* A、[递推关系](javascript:void(0);)
* B、[自顶向下计算](javascript:void(0);)
* C、[从大到小计算](javascript:void(0);)
* D、[子问题重叠的计算](javascript:void(0);)

窗体底端

正确答案： D

72

对于含有n个元素的排列树问题,最坏情况下其解空间的叶结点数目为( )。

窗体顶端

* A、[n!](javascript:void(0);)
* B、[[QQ截图20190624100201.png](javascript:void(0);)](javascript:void(0);)
* C、[[QQ截图20190624100223.png](javascript:void(0);)](javascript:void(0);)
* D、[[http://p.ananas.chaoxing.com/star3/origin/bb56b54898552fb70ebd58a9f422143e.png](javascript:void(0);)](javascript:void(0);)

窗体底端

正确答案： A

73

剪枝函数包括（ ）和约束函数。

窗体顶端

* A、[启发式函数](javascript:void(0);)
* B、[限界函数](javascript:void(0);)
* C、[估计函数](javascript:void(0);)
* D、[最优函数](javascript:void(0);)

窗体底端

正确答案： B

74

改进分治算法的方法有（ ）和改进划分的对称性。

窗体顶端

* A、[减少子问题数](javascript:void(0);)
* B、[备忘录](javascript:void(0);)
* C、[拟阵原理](javascript:void(0);)
* D、[加速原理](javascript:void(0);)

窗体底端

正确答案： A

75

待排序文件基本有序时,下面哪种排序方法,效率最差?

窗体顶端

* A、[堆排序](javascript:void(0);)
* B、[快速排序](javascript:void(0);)
* C、[冒泡排序](javascript:void(0);)
* D、[归并排序](javascript:void(0);)

窗体底端

正确答案： B

76

原问题的最优解包含其子问题的最优解,这是()性质

窗体顶端

* A、[贪心选择](javascript:void(0);)
* B、[无后效性](javascript:void(0);)
* C、[最优子结构](javascript:void(0);)
* D、[重叠子问题](javascript:void(0);)

窗体底端

正确答案： B

77

下面描述错误的是()

窗体顶端

* A、[求解问题的输入量,称为问题的规模.](javascript:void(0);)
* B、[时间复杂度是输入规模n的函数。](javascript:void(0);)
* C、[时间复杂度衡量算法的效率。](javascript:void(0);)
* D、[空间复杂度是算法执行所需所有空间的资源量。](javascript:void(0);)

窗体底端

正确答案： D

78

两个n/2长度的有序数组合并为新的有序数组的时间为()

窗体顶端

* A、[[QQ截图20190624100847.png](javascript:void(0);)](javascript:void(0);)
* B、[nlogn](javascript:void(0);)
* C、[n](javascript:void(0);)
* D、[n/2](javascript:void(0);)

窗体底端

正确答案： C

79

分治算法与动态规划算法的不同点是()

窗体顶端

* A、[递推关系](javascript:void(0);)
* B、[子问题独立](javascript:void(0);)
* C、[最优子结构](javascript:void(0);)
* D、[小问题易求解](javascript:void(0);)

窗体底端

正确答案： B

80

回溯法和分支限界法的主要区别在于,回溯法求取( ).

窗体顶端

* A、[一个解](javascript:void(0);)
* B、[极大解](javascript:void(0);)
* C、[极小解](javascript:void(0);)
* D、[一个解或所有解](javascript:void(0);)

窗体底端

正确答案： D

81

下列算法中不能解决0/1背包问题的是（）

窗体顶端

* A、[贪心法](javascript:void(0);)
* B、[动态规划](javascript:void(0);)
* C、[回溯法](javascript:void(0);)
* D、[分支限界法](javascript:void(0);)

窗体底端

正确答案： A

82

SPFA算法的时间复杂度为O（）

窗体顶端

* A、[mn](javascript:void(0);)
* B、[m+nlgn](javascript:void(0);)
* C、[mlogn](javascript:void(0);)
* D、[n](javascript:void(0);)[2](javascript:void(0);)

窗体底端

正确答案： A

83

下面关于贪心算法的说法错误的是()

窗体顶端

* A、[贪心算法的思想是寻求局部最优解,逐步达到全局最优解](javascript:void(0);)
* B、[贪心算法总能找到可行解,但未必是最优解。](javascript:void(0);)
* C、[贪心算法的思想是依据贪婪准则作出决策,逐步构造解值。](javascript:void(0);)
* D、[未来不影响过去指的是无后效性的性质。](javascript:void(0);)

窗体底端

正确答案： D

84

下面不是动态规划算法的基本要素的是（ ）。

窗体顶端

* A、[无后效性](javascript:void(0);)
* B、[独立子问题性质](javascript:void(0);)
* C、[最优子结构性质](javascript:void(0);)
* D、[重叠子问题性质](javascript:void(0);)

窗体底端

正确答案： B

85

给定n个元素,使用分块查找一般设分块的长度()

窗体顶端

* A、[n/3](javascript:void(0);)
* B、[[QQ截图20190624094834.png](javascript:void(0);)](javascript:void(0);)
* C、[[QQ截图20190624094902.png](javascript:void(0);)](javascript:void(0);)
* D、[n/2](javascript:void(0);)

窗体底端

正确答案： C

86

解决问题的基本步骤是（）。（1）算法设计（2）算法实现（3）数学建模（4）算法分析（5）正确性证明

窗体顶端

* A、[(3)(1)(4)(5)(2)](javascript:void(0);)
* B、[(3)(4)(1)(5)(2)](javascript:void(0);)
* C、[(3)(1)(5)(4)(2)](javascript:void(0);)
* D、[(1)(2)(3)(4)(5)](javascript:void(0);)

窗体底端

正确答案： C

87

()生成子集,便于实现集合的操作。

窗体顶端

* A、[增量构造法](javascript:void(0);)
* B、[二进制法](javascript:void(0);)
* C、[位向量法](javascript:void(0);)
* D、[法向量法](javascript:void(0);)

窗体底端

正确答案： B

88

对于稠密图，使用（）算法计算MST更适合

窗体顶端

* A、[Kruskal](javascript:void(0);)
* B、[Prim](javascript:void(0);)

窗体底端

正确答案： B

89

获得解不一定是正确解的算法是()。

窗体顶端

* A、[蒙特卡罗算法](javascript:void(0);)
* B、[拉斯维加斯算法](javascript:void(0);)
* C、[舍伍德算法](javascript:void(0);)
* D、[数值随机算法](javascript:void(0);)

窗体底端

正确答案： A

90

下面不是以空间换时间的方法有()

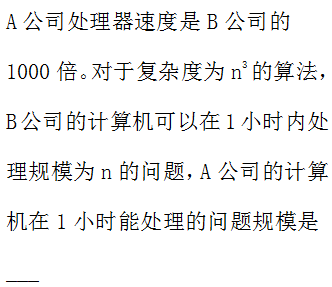
窗体顶端

* A、[预处理](javascript:void(0);)
* B、[预构造](javascript:void(0);)
* C、[动态规划](javascript:void(0);)
* D、[数据压缩](javascript:void(0);)

窗体底端

正确答案： D

91



（1.0分）

**0.0** 分

窗体顶端

* A、[n](javascript:void(0);)
* B、[10n](javascript:void(0);)
* C、[100n](javascript:void(0);)
* D、[[QQ截图20190624094621.png](javascript:void(0);)](javascript:void(0);)

窗体底端

正确答案： B

92

下面关于时间复杂度的描述错误的是()

窗体顶端

* A、[时间复杂度是最复杂部分的运行时间](javascript:void(0);)
* B、[时间复杂度是关键操作的运行时间](javascript:void(0);)
* C、[时间复杂度是在最坏情况下运行时间](javascript:void(0);)
* D、[时间复杂度是在平均情况下的运行时间](javascript:void(0);)

窗体底端

正确答案： D

93

QQ截图20190624102720.png

窗体顶端

* A、[100](javascript:void(0);)
* B、[10](javascript:void(0);)
* C、[1000](javascript:void(0);)
* D、[[QQ截图20190624102738.png](javascript:void(0);)](javascript:void(0);)

窗体底端

正确答案： A

94

下面有关递归与递推的说法错误的是()

窗体顶端

* A、[递归是逆向的,从大规模的问题逐步到小规模间题。](javascript:void(0);)
* B、[递推是正向的,从小规模的问题推解出大规模间题。](javascript:void(0);)
* C、[递归表现为自己调用自己,递推则没有这样的形式。](javascript:void(0);)
* D、[一般来说,递归的效率高于递推](javascript:void(0);)

窗体底端

正确答案： D

95

f(n)= Ω(g(n)),则 g(n)为f(n)的()

窗体顶端

* A、[上界](javascript:void(0);)
* B、[下界](javascript:void(0);)
* C、[同阶](javascript:void(0);)
* D、[低阶](javascript:void(0);)

窗体底端

正确答案： B

96

设有5000个无序的元素,希望用最快的速度挑选出其中前500个最大的元素,最好选用( )法。

窗体顶端

* A、[冒泡排序](javascript:void(0);)
* B、[快速排序](javascript:void(0);)
* C、[堆排序](javascript:void(0);)
* D、[基数排序](javascript:void(0);)

窗体底端

正确答案： C

97

实现二分搜索利用的算法是( )。

窗体顶端

* A、[分治策略](javascript:void(0);)
* B、[动态规划法](javascript:void(0);)
* C、[贪心法](javascript:void(0);)
* D、[回溯法](javascript:void(0);)

窗体底端

正确答案： A

98

从资源划分，算法的复杂度分为（ ）和（）。

窗体顶端

* A、[时间复杂度  空间复杂度](javascript:void(0);)
* B、[空间复杂度   平均复杂度](javascript:void(0);)
* C、[最好复杂度 最坏复杂度](javascript:void(0);)
* D、[时间复杂度   平均复杂度](javascript:void(0);)
* E、[间间复杂度   平均复杂度](javascript:void(0);)

窗体底端

正确答案： A

99

算法与程序的区别是（）

窗体顶端

* A、[输入](javascript:void(0);)
* B、[输出](javascript:void(0);)
* C、[确定性](javascript:void(0);)
* D、[有穷性](javascript:void(0);)

窗体底端

正确答案： D

100

FIFO是（  ）的搜索方式。

窗体顶端

* A、[回溯](javascript:void(0);)
* B、[分支限界](javascript:void(0);)
* C、[动态规划](javascript:void(0);)
* D、[贪心](javascript:void(0);)

窗体底端

正确答案： B

101

分支限界法解0-1背包问题时的解空间树是( )。

窗体顶端

* A、[子集树](javascript:void(0);)
* B、[排列树](javascript:void(0);)
* C、[深度优先生成树](javascript:void(0);)
* D、[广度优先生成树](javascript:void(0);)

窗体底端

正确答案： A

102

下面关于贪心算法错误的是()

窗体顶端

* A、[贪心算法总能找到可行解,并且是最优解。](javascript:void(0);)
* B、[问题的最优子结构性质是该问题可用贪心或动态规划算法求解的关键特征。](javascript:void(0);)
* C、[贪心算法一般预处理后再进行最优化选择。](javascript:void(0);)
* D、[贪心选择中每一步的局部最优解都构成全局最优解的一部分](javascript:void(0);)

窗体底端

正确答案： A

103重复

肯定获得可行解，但不一定是正确解的算法是（）。

窗体顶端

* A、[蒙特卡罗算法](javascript:void(0);)
* B、[拉斯维加斯算法舍伍德算法](javascript:void(0);)
* C、[舍伍德算法](javascript:void(0);)
* D、[数值随机算法](javascript:void(0);)

窗体底端

正确答案： A

104

备忘录与动态规划算法的不同点是()

窗体顶端

* A、[递推关系](javascript:void(0);)
* B、[自顶向下计算](javascript:void(0);)
* C、[子问题重叠](javascript:void(0);)
* D、[最优子结构](javascript:void(0);)

窗体底端

正确答案： B

105

备忘录方法使用( )的递归方式。

窗体顶端

* A、[自顶向下](javascript:void(0);)
* B、[自高到低](javascript:void(0);)
* C、[自低到高](javascript:void(0);)
* D、[自底向上](javascript:void(0);)

窗体底端

正确答案： A

106

20190603175445.png

窗体顶端

* A、[一般预流推进算法](javascript:void(0);)
* B、[先进先出预流推进算法](javascript:void(0);)
* C、[最高标号预流推进算法](javascript:void(0);)
* D、[最短增广路算法](javascript:void(0);)

窗体底端

正确答案： C ：

107

f(n)=100        n为奇数

f(n)=5n2+3n  n为偶数

则f(n)的下界为

窗体顶端

* A、[n](javascript:void(0);)
* B、[n](javascript:void(0);)[2](javascript:void(0);)
* C、[2](javascript:void(0);)[n](javascript:void(0);)
* D、[1](javascript:void(0);)

窗体底端

正确答案： D

108

下面不是动态规划的基本方法有（）。

窗体顶端

* A、[多重选择](javascript:void(0);)
* B、[增加变量](javascript:void(0);)
* C、[舍入](javascript:void(0);)
* D、[区间变量](javascript:void(0);)

窗体底端

正确答案： C

109

装载问题的回溯算法所需的计算时间为（        ）

窗体顶端

* A、[2](javascript:void(0);)[n](javascript:void(0);)
* B、[nlogn](javascript:void(0);)
* C、[n](javascript:void(0);)[2](javascript:void(0);)
* D、[n](javascript:void(0);)

窗体底端

正确答案： A

110

动态规划算法的基本要素有（ ）和最优子结构性质。

窗体顶端

* A、[分解合并性质](javascript:void(0);)
* B、[独立子问题性质](javascript:void(0);)
* C、[贪心选择性质](javascript:void(0);)
* D、[重叠子问题性质](javascript:void(0);)

窗体底端

正确答案： D

111类似

分块查找256个元素的数组,分成()块最好?

窗体顶端

* A、[8](javascript:void(0);)
* B、[16](javascript:void(0);)
* C、[32](javascript:void(0);)
* D、[64](javascript:void(0);)

窗体底端

正确答案： B

112

以下不可以使用分治法求解的是（  ）。

窗体顶端

* A、[线性选择问题](javascript:void(0);)
* B、[归并排序](javascript:void(0);)
* C、[0/1背包问题](javascript:void(0);)
* D、[棋盘覆盖问题](javascript:void(0);)

窗体底端

正确答案： C

113

对近似递增序列的线性表从小到大排序，使用哪种方法好？

窗体顶端

* A、[堆排序](javascript:void(0);)
* B、[快速排序](javascript:void(0);)
* C、[插入排序](javascript:void(0);)
* D、[归并排序](javascript:void(0);)

窗体底端

正确答案： C

114

分支限界法与回溯法都是在问题的解空间树T上搜索问题的解,二者()。

窗体顶端

* A、[求解目标不同,搜索方式相同](javascript:void(0);)
* B、[求解目标不同,搜索方式也不同](javascript:void(0);)
* C、[求解目标相同,搜索方式不同](javascript:void(0);)
* D、[求解目标相同,搜索方式也相同](javascript:void(0);)

窗体底端

正确答案： B

115

顺序查找长度为n的线性表,平均查找长度为()

窗体顶端

* A、[n](javascript:void(0);)
* B、[n/2](javascript:void(0);)
* C、[(n+1)/2](javascript:void(0);)
* D、[(n-1)/2](javascript:void(0);)

窗体底端

正确答案： C

116

( )是贪心算法与动态规划算法的共同点

窗体顶端

* A、[重叠子问题](javascript:void(0);)
* B、[构造最优解](javascript:void(0);)
* C、[贪心选择性质](javascript:void(0);)
* D、[最优子结构性质](javascript:void(0);)

窗体底端

正确答案： D

117

下面程序的时间复杂度为（）

x=1

for i=1 to n  do

                                for j=1 to i do

for k=1 to j do

                          x++

窗体顶端

* A、[O(n)](javascript:void(0);)
* B、[O(n](javascript:void(0);)[3](javascript:void(0);)[)](javascript:void(0);)
* C、[O(n](javascript:void(0);)[2](javascript:void(0);)[)](javascript:void(0);)
* D、[O(nlogn)](javascript:void(0);)

窗体底端

正确答案： B

118

插入排序的时间复杂度是()。

窗体顶端

* A、[O(n)](javascript:void(0);)
* B、[[QQ截图20190624104339.png](javascript:void(0);)](javascript:void(0);)
* C、[[QQ截图20190624104407.png](javascript:void(0);)](javascript:void(0);)
* D、[O(nlogn)](javascript:void(0);)

窗体底端

正确答案： C

119重复

算法复杂度分析的两种方法是()

窗体顶端

* A、[事前分析和事后统计](javascript:void(0);)
* B、[最坏情况的复杂度和最好情况的复杂度](javascript:void(0);)
* C、[最坏情况的复杂度和平均复杂度](javascript:void(0);)
* D、[最好情况的复杂度和平均复杂度](javascript:void(0);)

窗体底端

正确答案： A

120

待排序文件基本有序时,下面哪种排序方法,效率最高?

窗体顶端

* A、[堆排序](javascript:void(0);)
* B、[快速排序](javascript:void(0);)
* C、[冒泡排序](javascript:void(0);)
* D、[归并排序](javascript:void(0);)

窗体底端

正确答案： C

121

下面程序的时间复杂度为()

i=1

while(i<=n) do

 i=i\*2

窗体顶端

* A、[θ(n)](javascript:void(0);)
* B、[O(logn)](javascript:void(0);)
* C、[Ω(n^2)](javascript:void(0);)
* D、[o(logn)](javascript:void(0);)

窗体底端

正确答案： B

122

分支限界法解旅行商问题时的解空间树是（       ）。

窗体顶端

* A、[子集树](javascript:void(0);)
* B、[排列树](javascript:void(0);)
* C、[深度优先生成树](javascript:void(0);)
* D、[广度优先生成树](javascript:void(0);)

窗体底端

正确答案： B

123重复

下面不是以空间换时间的方法有()

窗体顶端

* A、[预处理](javascript:void(0);)
* B、[预构造](javascript:void(0);)
* C、[动态规划](javascript:void(0);)
* D、[数据压缩](javascript:void(0);)

窗体底端

正确答案： D

124

给定图G=(V,E), |V|=n, |E|=m, 其邻接表的空间复杂度为( )

窗体顶端

* A、[θ(m+n)](javascript:void(0);)
* B、[θ(n)](javascript:void(0);)
* C、[θ(mn)](javascript:void(0);)
* D、[θ(n^2)](javascript:void(0);)

窗体底端

正确答案： A

125

问题变换的目的有（）。(1)复杂变简单 (2)未知变已知 (3)隐式变显式 (4)难解变易解 （5）以上都是。

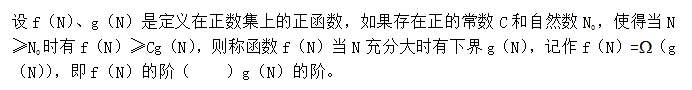
窗体顶端

* A、[(1)](javascript:void(0);)
* B、[(2)](javascript:void(0);)
* C、[(3)](javascript:void(0);)
* D、[(4)](javascript:void(0);)
* E、[(5)](javascript:void(0);)

窗体底端

正确答案： E

126



窗体顶端

* A、[不高于](javascript:void(0);)
* B、[不低于](javascript:void(0);)
* C、[等价于](javascript:void(0);)
* D、[逼近](javascript:void(0);)

窗体底端

正确答案： B

127

优先队列式分支限界法选取扩展结点的原则是（ ）

窗体顶端

* A、[先进先出](javascript:void(0);)
* B、[后进先出](javascript:void(0);)
* C、[结点的优先级](javascript:void(0);)
* D、[随机](javascript:void(0);)

窗体底端

正确答案： C

128

可能获得解,且一定是准确解的算法是()。

窗体顶端

* A、[蒙特卡罗算法](javascript:void(0);)
* B、[拉斯维加斯算法](javascript:void(0);)
* C、[舍伍德算法](javascript:void(0);)
* D、[数值随机算法](javascript:void(0);)

窗体底端

正确答案： B

129

从长度为n的数组中多次查找数据,使用( )方法好。

窗体顶端

* A、[顺序查找](javascript:void(0);)
* B、[随机查找](javascript:void(0);)
* C、[排序后折半查找](javascript:void(0);)
* D、[无序查找](javascript:void(0);)

窗体底端

正确答案： C

130

下面不是动态规划算法特点的是()

窗体顶端

* A、[自底向上计算](javascript:void(0);)
* B、[无后效性](javascript:void(0);)
* C、[子问题独立](javascript:void(0);)
* D、[子问题重叠](javascript:void(0);)

窗体底端

正确答案： C

131

求解高阶递推方程一般使用（）迭代方法

窗体顶端

* A、[差消迭代](javascript:void(0);)
* B、[换元迭代](javascript:void(0);)
* C、[直接迭代](javascript:void(0);)

窗体底端

正确答案： A

132

假设算法A的计算时间为T(n)=2^n, 在计算机A上输入规模为n时算法A的运行时间为t秒。计算机B的运行速度是A的64倍,在t秒时间计算机B运行算法A的输入规模是().

窗体顶端

* A、[n+6](javascript:void(0);)
* B、[64n](javascript:void(0);)
* C、[6n](javascript:void(0);)
* D、[[QQ截图20190624102831.png](javascript:void(0);)](javascript:void(0);)

窗体底端

正确答案： A

133

以下是NP完全问题的（）

窗体顶端

* A、[0-1背包](javascript:void(0);)
* B、[顶点覆盖](javascript:void(0);)
* C、[最短路](javascript:void(0);)
* D、[最大公因子](javascript:void(0);)

窗体底端

正确答案： A

134

NP类问题可以( )。

窗体顶端

* A、[多项式时间计算](javascript:void(0);)
* B、[指数时间验证](javascript:void(0);)
* C、[多项式时间验证](javascript:void(0);)
* D、[指数时间计算](javascript:void(0);)

窗体底端

正确答案： C

135

主方法可以求解满足T(n)=aT(n/b) + f (n) 形式的递推方程， 则下列关于方程中的约束中不准确的是？

窗体顶端

* A、[对于系数a，必须满足a>=1](javascript:void(0);)
* B、[对于系数b，必须满足b>1](javascript:void(0);)
* C、[若对于常数ε>0，f(n)=O(n](javascript:void(0);)[log](javascript:void(0);)[b](javascript:void(0);)[a-ε](javascript:void(0);)[)，则T(n)=Θ(n](javascript:void(0);)[log](javascript:void(0);)[b](javascript:void(0);)[a](javascript:void(0);)[)](javascript:void(0);)
* D、[若f(n)=O(nlogba)，则T(n)=](javascript:void(0);)Θ(nlogbalogn)

窗体底端

正确答案： D

136

一个问题可用动态规划算法或贪心算法求解的关键特征是问题的( )。

窗体顶端

* A、[重叠子问题](javascript:void(0);)
* B、[最优子结构性质](javascript:void(0);)
* C、[贪心选择性质](javascript:void(0);)
* D、[定义最优解](javascript:void(0);)

窗体底端

正确答案： B

**二、多选题 （题数：75，考试抽取5道，分值10分）**

1

通过降低子问题合并时间,降低分治算法时间复杂度的有()

窗体顶端

* A、[大整数乘法](javascript:void(0);)
* B、[计数逆序](javascript:void(0);)
* C、[线性时间选择](javascript:void(0);)
* D、[最接近点对](javascript:void(0);)

窗体底端

正确答案： BD

2

子集生成方法有()

窗体顶端

* A、[增量构造法](javascript:void(0);)
* B、[二进制法](javascript:void(0);)
* C、[位向量法](javascript:void(0);)
* D、[法向量法](javascript:void(0);)

窗体底端

正确答案： ABC

3

下面时间复杂度的描述正确的是()

窗体顶端

* A、[logn!=θ( nlogn )](javascript:void(0);)
* B、[[QQ截图20190624093005.png](javascript:void(0);)](javascript:void(0);)
* C、[折半查找的时间复杂度为θ(nlogn)](javascript:void(0);)
* D、[从n个数中查找最小值的时间复杂度为O(n)](javascript:void(0);)

窗体底端

正确答案： ABD 我的答案：

4

求解递推方程的迭代法分为（   ）

窗体顶端

* A、[直接迭代](javascript:void(0);)
* B、[差消迭代](javascript:void(0);)
* C、[换元迭代](javascript:void(0);)
* D、[主定理](javascript:void(0);)

窗体底端

正确答案： ABC

5

贪心算法的基本要素是()

窗体顶端

* A、[贪心选择的性质](javascript:void(0);)
* B、[无后效性性质](javascript:void(0);)
* C、[最优子结构性质](javascript:void(0);)
* D、[独立子问题的性质](javascript:void(0);)

窗体底端

正确答案： ABC

6

给定网络 N=(V, E)的一个流 f,f需满足的条件是

窗体顶端

* A、

[[QQ截图20190624104503.png](javascript:void(0);)](javascript:void(0);)

* B、

[[QQ截图20190624104519.png](javascript:void(0);)](javascript:void(0);)

* C、[源点s的流出量=|f|](javascript:void(0);)
* D、[汇点t的流入量=|f|](javascript:void(0);)

窗体底端

正确答案： ABCD

7

递归函数的要素是（）

窗体顶端

* A、[边界条件](javascript:void(0);)
* B、[递归方程](javascript:void(0);)
* C、[输入](javascript:void(0);)
* D、[迭代](javascript:void(0);)

窗体底端

正确答案： AB

8

P类问题可以( )。

窗体顶端

* A、[多项式时间计算](javascript:void(0);)
* B、[指数时间计算](javascript:void(0);)
* C、[指数时间验证](javascript:void(0);)
* D、[多项式时间验证](javascript:void(0);)

窗体底端

正确答案： AD

9

下面属于NP完全问题的是（）

窗体顶端

* A、[SAT](javascript:void(0);)
* B、[最大独立集](javascript:void(0);)
* C、[最小顶点覆盖](javascript:void(0);)
* D、[旅行商问题](javascript:void(0);)

窗体底端

正确答案： ABCD

10

关于带需求的流通下面说法正确的是()。f(e)为边e的流量,c(e)为边e的容量。

窗体顶端

* A、

[[QQ截图20190624093116.png](javascript:void(0);)](javascript:void(0);)

* B、[对任意顶点v-{s,t},顶点的净流量=0](javascript:void(0);)
* C、[供给和=需求和](javascript:void(0);)
* D、

[[QQ截图20190624093139.png](javascript:void(0);)](javascript:void(0);)

窗体底端

正确答案： AC

11

下面公式正确的是()

窗体顶端

* A、

[[QQ截图20190624103202.png](javascript:void(0);)](javascript:void(0);)

* B、

[[QQ截图20190624103217.png](javascript:void(0);)](javascript:void(0);)

* C、

[[QQ截图20190624103230.png](javascript:void(0);)](javascript:void(0);)

* D、

[[QQ截图20190624103246.png](javascript:void(0);)](javascript:void(0);)

窗体底端

正确答案： ABC

12

下面那些算法的时间复杂度为O（n2）

窗体顶端

* A、[顺序查找](javascript:void(0);)
* B、[折半查找](javascript:void(0);)
* C、[插入排序](javascript:void(0);)
* D、[冒泡排序](javascript:void(0);)
* E、[折半插入排序](javascript:void(0);)

窗体底端

正确答案： CDE

13

回溯法的效率依赖于下列哪些因素（   ）

窗体顶端

* A、[满足显约束的值的个数](javascript:void(0);)
* B、[计算约束函数的时间](javascript:void(0);)
* C、[计算限界函数的时间](javascript:void(0);)
* D、[确定解空间的时间](javascript:void(0);)

窗体底端

正确答案： ABC

14

最短路算法中适用于负权图的是()

窗体顶端

* A、[Floyd算法](javascript:void(0);)
* B、[SPFA算法](javascript:void(0);)
* C、[Bellman算法](javascript:void(0);)
* D、[Dijkstra算法](javascript:void(0);)

窗体底端

正确答案： ABC

15

动态规划算法的特点()

窗体顶端

* A、[自底向上计算](javascript:void(0);)
* B、[自顶向下计算](javascript:void(0);)
* C、[子问题独立](javascript:void(0);)
* D、[子问题重叠](javascript:void(0);)

窗体底端

正确答案： AD

16

区间动态规划的计算次序是()

窗体顶端

* A、[先小区间后大区间](javascript:void(0);)
* B、[先大区间后小区间](javascript:void(0);)
* C、[自底向上](javascript:void(0);)
* D、[自顶向下](javascript:void(0);)

窗体底端

正确答案： AC

17

问题变换的方法有（ ）

窗体顶端

* A、[实例简单化](javascript:void(0);)
* B、[问题约简](javascript:void(0);)
* C、[表达变换](javascript:void(0);)
* D、[分支](javascript:void(0);)

窗体底端

正确答案： ABC

18

下列哪个结点属于回溯法的结点类型？（     ）

窗体顶端

* A、[扩展结点](javascript:void(0);)
* B、[活结点](javascript:void(0);)
* C、[根结点](javascript:void(0);)
* D、[死结点](javascript:void(0);)

窗体底端

正确答案： ABD

19

 T(n) = T(n-1) + n ，T(1)=1，则 T(n) =（）

窗体顶端

* A、[θ(n](javascript:void(0);)[2](javascript:void(0);)[)](javascript:void(0);)
* B、[n(n+1)/2](javascript:void(0);)
* C、[O(n](javascript:void(0);)[2](javascript:void(0);)[)](javascript:void(0);)
* D、[Ω(n](javascript:void(0);)[2](javascript:void(0);)[)](javascript:void(0);)

窗体底端

正确答案： ABCD

20

最好情况下,时间复杂度为O(n)的排序算法有()

窗体顶端

* A、[冒泡排序](javascript:void(0);)
* B、[计数排序](javascript:void(0);)
* C、[插入排序](javascript:void(0);)
* D、[直接选择排序](javascript:void(0);)

窗体底端

正确答案： ABC

21

改进分治算法的方法有()

窗体顶端

* A、[减少问题的规模](javascript:void(0);)
* B、[改进分治的均衡度](javascript:void(0);)
* C、[减少合并的时间](javascript:void(0);)
* D、[减少子问题的个数](javascript:void(0);)

窗体底端

正确答案： BCD

22

从活结点表中选择下一个扩展结点的不同方式将导致不同的分支限界法,最常见的方式有( )。

窗体顶端

* A、[队列式分支限界法](javascript:void(0);)
* B、[优先队列式分支限界法](javascript:void(0);)
* C、[栈式分支限界法](javascript:void(0);)
* D、[FIFO分支限界法](javascript:void(0);)

窗体底端

正确答案： ABD

23

下面说法正确的是（）

窗体顶端

* A、[使用限界函数作优先级, 第一个加入队列的叶子就是最优解](javascript:void(0);)
* B、[用约束函数在扩展结点处剪去不满足约束的子树；](javascript:void(0);)
* C、[用限界函数剪去得不到最优解的子树。](javascript:void(0);)
* D、[回溯和分支限界都是动态生成解空间树。](javascript:void(0);)

窗体底端

正确答案： BCD

24

以下关于判定问题难易处理的叙述中错误的是（    ）。

窗体顶端

* A、[可以由多项式时间算法求解的问题是难处理的](javascript:void(0);)
* B、[需要超过多项式时间算法求解的问题是易处理的](javascript:void(0);)
* C、[可以由多项式时间算法求解的问题是易处理的](javascript:void(0);)
* D、[需要超过多项式时间算法求解的问题是不能处理的](javascript:void(0);)

窗体底端

正确答案： ABD

25

最小生成树问题可以使用的算法有（ ）

窗体顶端

* A、[Kruskal](javascript:void(0);)
* B、[Prim](javascript:void(0);)
* C、[Solim](javascript:void(0);)
* D、[Dijkstra](javascript:void(0);)

窗体底端

正确答案： ABC

26

分治算法与动态规划算法的相同点是（）

窗体顶端

* A、[递推关系](javascript:void(0);)
* B、[子问题独立](javascript:void(0);)
* C、[子问题重叠](javascript:void(0);)
* D、[最优子结构](javascript:void(0);)

窗体底端

正确答案： AD

27

N后问题利用( )减少搜索。

窗体顶端

* A、[对称性](javascript:void(0);)
* B、[约束函数](javascript:void(0);)
* C、[重排原理](javascript:void(0);)
* D、[加速原理](javascript:void(0);)

窗体底端

正确答案： ABC

28

备忘录算法的特点（）

窗体顶端

* A、[自底向上计算](javascript:void(0);)
* B、[自顶向下计算](javascript:void(0);)
* C、[从大到小计算](javascript:void(0);)
* D、[从小到大计算](javascript:void(0);)

窗体底端

正确答案： BC

29

下列算法中能解决0/1背包问题的是()

窗体顶端

* A、[贪心法](javascript:void(0);)
* B、[动态规划](javascript:void(0);)
* C、[回溯法](javascript:void(0);)
* D、[分支限界法](javascript:void(0);)

窗体底端

正确答案： BCD

30

下面分治算法的说法正确的是()

窗体顶端

* A、[处理随机排列的数组时,合并排序比快速排序快。](javascript:void(0);)
* B、[三分法的判定树是三叉树](javascript:void(0);)
* C、[最小堆中每个元素调整的次数不超过树高 θ(logn)。](javascript:void(0);)
* D、[二分法子问题不独立的情况可以使用分治算法计算,但计算量大](javascript:void(0);)

窗体底端

正确答案： BCD

31

最大独立集问题和（）问题等价。

窗体顶端

* A、[最大团](javascript:void(0);)
* B、[最小顶点覆盖](javascript:void(0);)
* C、[区间调度问题](javascript:void(0);)
* D、[稳定匹配问题](javascript:void(0);)

窗体底端

正确答案： AB

32

下面关于NP问题说法错误的是( )

窗体顶端

* A、[NP问题都是不可能解决的问题](javascript:void(0);)
* B、[P类问题包含在NP类问题中](javascript:void(0);)
* C、[NP完全问题是P类问题的子集](javascript:void(0);)
* D、[NP类问题包含在P类问题中](javascript:void(0);)

窗体底端

正确答案： ACD

33

通过减少子问题个数，降低分治算法时间复杂度的有（）

窗体顶端

* A、[大整数乘法](javascript:void(0);)
* B、[Strassen矩阵乘法](javascript:void(0);)
* C、[线性时间选择](javascript:void(0);)
* D、[最接近点对](javascript:void(0);)

窗体底端

正确答案： AB

34

回溯算法的效率在很大程度上依赖的因素有()

窗体顶端

* A、[产生x[k]的时间。](javascript:void(0);)
* B、[满足显约束的x[k]值的个数。](javascript:void(0);)
* C、[计算可行性约束函数constraint和 上界函数bound的时间。](javascript:void(0);)
* D、[满足可行性约束函数和上界函数的所有x[k]的个数。](javascript:void(0);)

窗体底端

正确答案： ABCD

35

给定两张喜欢列表，稳定匹配问题的输出是（  ） 。

窗体顶端

* A、[完美匹配](javascript:void(0);)
* B、[没有不稳定配对](javascript:void(0);)
* C、[最大匹配](javascript:void(0);)
* D、[稳定匹配](javascript:void(0);)

窗体底端

正确答案： ABCD

36

分数拆分问题的枚举算法通过（）方法进行了优化。

窗体顶端

* A、[减少枚举变量](javascript:void(0);)
* B、[减少枚举变量的值域](javascript:void(0);)
* C、[优化数据结构](javascript:void(0);)
* D、[优化数学模型](javascript:void(0);)

窗体底端

正确答案： ABD

37

时间复杂度为O(nlogn)的排序算法有()

窗体顶端

* A、[堆排序](javascript:void(0);)
* B、[快速排序](javascript:void(0);)
* C、[合并排序](javascript:void(0);)
* D、[计数排序](javascript:void(0);)

窗体底端

正确答案： AC

38

贪心算法的常用证明方法有()。

窗体顶端

* A、[领先](javascript:void(0);)
* B、[反证](javascript:void(0);)
* C、[界](javascript:void(0);)
* D、[交换论证](javascript:void(0);)

窗体底端

正确答案： ABCD

39

改进FF网络流算法，可以通过选择（  ）增广路，降低时间复杂度。

窗体顶端

* A、[最大容量](javascript:void(0);)
* B、[最短路径](javascript:void(0);)
* C、[最大瓶颈容量](javascript:void(0);)
* D、[边数最少](javascript:void(0);)

窗体底端

正确答案： ABCD

40

区间问题包含()

窗体顶端

* A、[区间调度](javascript:void(0);)
* B、[区间划分](javascript:void(0);)
* C、[区间选点](javascript:void(0);)
* D、[区间覆盖](javascript:void(0);)

窗体底端

正确答案： ABCD

41

递归变为非递归的方法有（）

窗体顶端

* A、[模拟栈](javascript:void(0);)
* B、[递推](javascript:void(0);)
* C、[尾递归](javascript:void(0);)
* D、[循环](javascript:void(0);)

窗体底端

正确答案： ABC

42

回溯法解题步骤：

窗体顶端

* A、[针对所给问题，定义问题的解空间](javascript:void(0);)
* B、[确定易于搜索的解空间结构](javascript:void(0);)
* C、[确定最优子结构的性质](javascript:void(0);)
* D、[以深度优先方式搜索解空间，在搜索过程中用剪枝函数避免无效搜索](javascript:void(0);)

窗体底端

正确答案： ABD 我的答案：

43

分治算法的适用条件有()

窗体顶端

* A、[问题可以分解为规模较小的子问题](javascript:void(0);)
* B、[小规模子问题可解](javascript:void(0);)
* C、[子问题可合并为问题的解](javascript:void(0);)
* D、[子问题相互独立](javascript:void(0);)

窗体底端

正确答案： ABCD

44

从资源划分,算法的复杂度分为( )。

窗体顶端

* A、[时间复杂度](javascript:void(0);)
* B、[空间复杂度](javascript:void(0);)
* C、[平均复杂度](javascript:void(0);)
* D、[平摊复杂度](javascript:void(0);)

窗体底端

正确答案： AB

45

提高事后统计方法准确度的有（）

窗体顶端

* A、[重复测试](javascript:void(0);)
* B、[加大n](javascript:void(0);)
* C、[典型实例测试](javascript:void(0);)
* D、[优化算法](javascript:void(0);)

窗体底端

正确答案： ABC

46

下面关于程序和算法的说法正确的是（）。

窗体顶端

* A、[算法的每一步骤必须要有确切的含义，必须是清楚的、无二义的。](javascript:void(0);)
* B、[程序是算法用某种程序设计语言的具体实现。](javascript:void(0);)
* C、[程序总是在有穷步的运算后终止。](javascript:void(0);)
* D、[算法是一个过程，计算机每次求解是针对问题的一个实例求解。](javascript:void(0);)

窗体底端

正确答案： ABD

47

递归一般用于解决问题有（）：

窗体顶端

* A、[数据的定义是按递归定义的。](javascript:void(0);)
* B、[问题解法按递归实现。（回溯）](javascript:void(0);)
* C、[数据的结构形式是按递归定义的。](javascript:void(0);)
* D、[迭代问题](javascript:void(0);)

窗体底端

正确答案： ABC

48

备忘录与递归算法的相同点是()

窗体顶端

* A、[递推关系](javascript:void(0);)
* B、[自顶向下计算](javascript:void(0);)
* C、[从大到小计算](javascript:void(0);)
* D、[子问题重叠](javascript:void(0);)

窗体底端

正确答案： AD

49

备忘录算法的特点()

窗体顶端

* A、[自底向上计算](javascript:void(0);)
* B、[自顶向下计算](javascript:void(0);)
* C、[子问题独立](javascript:void(0);)
* D、[子问题重叠](javascript:void(0);)

窗体底端

正确答案： BD

50

[分治法在每一层递归上有三个步骤（）](javascript:void(0);)

窗体顶端

* A、分解
* B、解决
* C、合并
* D、选择

窗体底端

正确答案： ABC

51

最短路算法中适用于稀疏图的是（）

窗体顶端

* A、[Floyd算法](javascript:void(0);)
* B、[SPFA算法](javascript:void(0);)
* C、[Bellman算法](javascript:void(0);)
* D、[Dijkstra算法](javascript:void(0);)

窗体底端

正确答案： BCD

52

对于NP完全问题,可采取的解题策略有()。

窗体顶端

* A、[用启发式方法求解](javascript:void(0);)
* B、[分支限界法求解小实例](javascript:void(0);)
* C、[用随机算法求解](javascript:void(0);)
* D、[只求近似解](javascript:void(0);)

窗体底端

正确答案： ABCD

53

（ ）肯定获得最优解。

窗体顶端

* A、[分支限界](javascript:void(0);)
* B、[贪心算法](javascript:void(0);)
* C、[随机算法](javascript:void(0);)
* D、[动态规划算法](javascript:void(0);)

窗体底端

正确答案： AD

54

问题的状态生成法有（）

窗体顶端

* A、[子集树生成法](javascript:void(0);)
* B、[深度优先生成法](javascript:void(0);)
* C、[宽度优先生成法](javascript:void(0);)
* D、[排列树生成法](javascript:void(0);)

窗体底端

正确答案： BC

55

( )肯定获得最优解。

窗体顶端

* A、[回溯算法](javascript:void(0);)
* B、[贪心算法](javascript:void(0);)
* C、[随机算法](javascript:void(0);)
* D、[枚举算法](javascript:void(0);)

窗体底端

正确答案： AD

56

属于最短路增广路算法的有()

窗体顶端

* A、[FF算法](javascript:void(0);)
* B、[EK算法](javascript:void(0);)
* C、[Dinic算法](javascript:void(0);)
* D、[ISAP算法](javascript:void(0);)

窗体底端

正确答案： BCD

57

下面说法正确的是（）

窗体顶端

* A、[现实计算机上无法产生真正的随机数](javascript:void(0);)
* B、[求解同一实例用同一随机化算法求解两次，所用时间和所得结果可能完全不同。](javascript:void(0);)
* C、[蒙特卡罗算法总是能提供问题的一个解,但可能给出错误解。](javascript:void(0);)
* D、[舍伍德算法的精髓不是避免最坏的情况，而是设法消除最坏情况和特定实例的关联性。](javascript:void(0);)

窗体底端

正确答案： ABCD

58重复

枚举算法的优化方法有()

窗体顶端

* A、[减少枚举变量](javascript:void(0);)
* B、[减少枚举变量的值域](javascript:void(0);)
* C、[优化数据结构](javascript:void(0);)
* D、[优化数学模型](javascript:void(0);)

窗体底端

正确答案： ABCD

59

动态规划算法的特点（）

窗体顶端

* A、[自底向上计算](javascript:void(0);)
* B、[自顶向下计算](javascript:void(0);)
* C、[从大到小计算](javascript:void(0);)
* D、[从小到大计算](javascript:void(0);)

窗体底端

正确答案： AD

60

20190603175546.png

窗体顶端

* A、[最大独立数](javascript:void(0);)
* B、[最大匹配数](javascript:void(0);)
* C、[最小顶点覆盖](javascript:void(0);)
* D、[最小边覆盖](javascript:void(0);)

窗体底端

正确答案： AD

61

问题变换的目的和方式有()。

窗体顶端

* A、[复杂变简单](javascript:void(0);)
* B、[未知变已知](javascript:void(0);)
* C、[难解变易解](javascript:void(0);)
* D、[隐式变显式](javascript:void(0);)

窗体底端

正确答案： ABCD

62

复杂度比较方法有( )

窗体顶端

* A、[对数](javascript:void(0);)
* B、[积分](javascript:void(0);)
* C、[极限](javascript:void(0);)
* D、[放大](javascript:void(0);)

窗体底端

正确答案： ABCD

63类似

给定二分图G = <V, E>中无孤立点,|V|=n,其最大流算法求得最大流f, 则 G的()=f.

窗体顶端

* A、[最大独立数](javascript:void(0);)
* B、[最大匹配数](javascript:void(0);)
* C、[最小顶点覆盖](javascript:void(0);)
* D、[最小边覆盖](javascript:void(0);)

窗体底端

正确答案： BC

64重复

分治法在每一层递归上有三个步骤（）

窗体顶端

* A、[分解](javascript:void(0);)
* B、[解决](javascript:void(0);)
* C、[合并](javascript:void(0);)
* D、[选择](javascript:void(0);)

窗体底端

正确答案： ABC

65

顺序查找适合的数据结构是（）

窗体顶端

* A、[散列存储](javascript:void(0);)
* B、[顺序存储](javascript:void(0);)
* C、[链式存储](javascript:void(0);)
* D、[压缩存储](javascript:void(0);)

窗体底端

正确答案： BC

66

下面说法正确的是（）

窗体顶端

* A、[随机算法是一种使用概率和统计方法在其执行过程中对于下一计算步骤作出随机选择的算法.](javascript:void(0);)
* B、[当最坏和平均情况差别较大时, 舍伍德算法可以消除好坏实例的差别，达到平均实例的性能.](javascript:void(0);)
* C、[线性同余法是产生伪随机数的最常用的方法](javascript:void(0);)
* D、[增加蒙特卡罗算法的求解次数, 可使求解错误的概率任意小。](javascript:void(0);)

窗体底端

正确答案： ABCD

67

下面分治算法的说法正确的是()

窗体顶端

* A、[分治法的设计思想是大事化小,各个击破,分而治之。](javascript:void(0);)
* B、[每次都将问题分解为原问题规模的一半进行求解,称为二分法。](javascript:void(0);)
* C、[分治法将原问题分解为若干个规模较小、相互独立、完全相同的子问题。](javascript:void(0);)
* D、[减治法是把一个问题转化成一个子问题来解决。](javascript:void(0);)

窗体底端

正确答案： ABD

68

下面时间复杂度的描述正确的是()

窗体顶端

* A、[logn!=θ( nlogn )](javascript:void(0);)
* B、[a^(logn) = n^(loga )](javascript:void(0);)
* C、[折半查找的时间复杂度为θ(nlogn)](javascript:void(0);)
* D、[从n个数中查找最小值的时间复杂度为O(n)](javascript:void(0);)

窗体底端

正确答案： ABD

69

分支限界法与回溯法的不同点是什么？

窗体顶端

* A、[求解目标不同](javascript:void(0);)
* B、[搜索方式不同](javascript:void(0);)
* C、[对扩展结点的扩展方式不同](javascript:void(0);)
* D、[存储空间的要求不同](javascript:void(0);)

窗体底端

正确答案： ABCD

70

用分支限界法设计算法的步骤是：

窗体顶端

* A、[针对所给问题，定义问题的解空间（对解进行编码）；](javascript:void(0);)
* B、[确定易于搜索的解空间结构（按树或图组织解） ；](javascript:void(0);)
* C、[定义最优子结构](javascript:void(0);)
* D、[以广度优先或以最小耗费（最大收益）优先的方式搜索解空间，并在搜索过程中用剪枝函数避免无效搜索。](javascript:void(0);)

窗体底端

正确答案： ABD

71

算法的性质有()

窗体顶端

* A、[输入](javascript:void(0);)
* B、[输出](javascript:void(0);)
* C、[确定性](javascript:void(0);)
* D、[有穷性](javascript:void(0);)

窗体底端

正确答案： ABCD

72

求解递推方程的方法有()

窗体顶端

* A、[迭代法](javascript:void(0);)
* B、[递归树](javascript:void(0);)
* C、[归纳法](javascript:void(0);)
* D、[主定理](javascript:void(0);)

窗体底端

正确答案： ABCD

73类似

动态规划算法的基本要素有( )。

窗体顶端

* A、[贪心选择性质](javascript:void(0);)
* B、[最优子结构性质](javascript:void(0);)
* C、[无后效性](javascript:void(0);)
* D、[重叠子问题性质](javascript:void(0);)

窗体底端

正确答案： BCD

74

下面说法正确的是()

窗体顶端

* A、[多源点和多汇点的网络流问题可以通过增加一个“超源点”和“超汇点”转化为单源点和单汇点的网络流问题](javascript:void(0);)
* B、[无向图的每条边变为方向相反的两条边,容量是原边的容量,这样无向图的最大流问题变换为有向图的最大流问题。](javascript:void(0);)
* C、[剩余网络中从源s到汇t的最小费用路是剩余网络中从s到t的以费用为权的最短路.](javascript:void(0);)
* D、[最小费用最大流算法寻找从源点s到汇点t的最小费用路,然后沿最小费用路增流,直至找到最小费用流。](javascript:void(0);)

窗体底端

正确答案： ABCD

75类似

时间复杂度为O(n^2)的排序算法有()

窗体顶端

* A、[冒泡排序](javascript:void(0);)
* B、[快速排序](javascript:void(0);)
* C、[插入排序](javascript:void(0);)
* D、[直接选择排序](javascript:void(0);)

窗体底端

正确答案： ABCD

**三、填空题 （题数：64，考试无填空，可作为选择判断的补充）**

1

回答是与否的问题是\_\_\_问题。

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

判定

窗体底端

2

20190603171140.png

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

10n

窗体底端

3

构造一个解使目标函数最大或最小的问题是\_\_问题。

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

优化

窗体底端

4

使用回溯法进行状态空间树裁剪分支时一般有两个标准：可行性约束函数和限界函数，装载问题和旅行商问题正好是两种不同的类型，其中同时使用可行性约束函数和限界函数的进行裁剪的是        ，只使用限界函数进行裁剪的是         。

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

装载问题

**第二空：**

旅行商问题

窗体底端

5

分析下列方程的上界O(\_\_)和下界W(\_\_).

T(n) = T(n-1) + 1, T(1)=1

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

n

**第二空：**

n

窗体底端

6

分析下列程序的上界O(\_\_)和下界Ω(\_\_).

p=1

for i=1 to n^2 do

  for j=1 to i do

     p=p+i

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

n^4

**第二空：**

n^4

窗体底端

7

QQ截图20190624103621.png

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

n^2

**第二空：**

n^2

窗体底端

8

按照霍纳法则，计算p(x) = anxn+ an-1xn-1+… + a1x1+ a0的数量级为\_\_\_\_ 。

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

n

窗体底端

9

背包问题，背包容量C=20 ，物品价值pi =[4, 8，15, 1, 6，3], 物品重量wi=[5, 3，2, 10, 4, 8].如果是部分背包问题，求装入背包的最大价值和相应装入物品。

该问题最好使用（\_\_\_）算法求解. 装入背包的最大价值是（\_\_\_\_\_），对应的完整物品的编号是（\_\_\_\_）、（\_\_\_\_）、（\_\_\_\_）、（\_\_\_）。

如果物品数为n，算法的时间复杂度为O（）。

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

贪心

**第二空：**

35.25

**第三空：**

1

**第四空：**

2

**第五空：**

3

**第六空：**

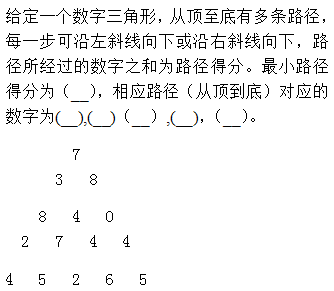
5

**第七空：**

nlogn

窗体底端

10



窗体顶端

[正确答案](javascript:void(0))

**第一空：**

20

**第二空：**

7

**第三空：**

3

**第四空：**

4

**第五空：**

4

**第六空：**

2

窗体底端

11

回溯法搜索解空间树时，常用的两种剪枝函数为     、     。

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

约束函数

**第二空：**

限界函数

我的答案:

窗体底端

12

多选式时间可验证问题是\_\_\_\_问题。

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

NP

窗体底端

13

矩阵 C中每一行选一个元素 ,使选择的元素不同列,并且元素之和最小。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Job 1 | Job 2 | Job 3 | Job 4 |
| Person a | 9 | 5 | 6 | 8 |
| Person b | 6 | 4 | 3 | 7 |
| Person c | 5 | 8 | 1 | 8 |
| Person d | 7 | 6 | 9 | 4 |

问题的解空间是(\_\_\_)树。最小元素和(\_\_\_\_),对应的安排是 a安排job(\_\_\_\_\_\_)、b安排job(\_\_\_)、c安排job(\_\_\_\_\_)、 d安排job(\_\_\_)。如果人数和任务数为n,时间复杂度是(\_\_\_\_)。

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

排列

**第二空：**

16

**第三空：**

2

**第四空：**

1

**第五空：**

3

**第六空：**

4

**第七空：**

n!

窗体底端

14

传教士和野人问题转换的图是\_\_\_\_。

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

状态空间图

窗体底端

15

插入排序最好情况下的时间复杂度为O(\_\_\_),最坏情况下的复杂度为O（\_\_\_）

窗体顶端

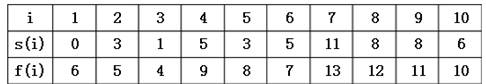
[正确答案](javascript:void(0))

**第一空：**n **第二空：**n2

窗体底端

16

某体育馆有一羽毛球场出租,现在总共有10位客户申请租用此羽毛球场,每个客户所租用的时间单元如下表所示,s(i)表示开始租用时刻,f(i)表示结束租用时刻,10个客户的申请如下表所示.同一时刻,该羽毛球场只能租借给一位客户,请设计一个租用安排方案,在这10位客户里面,使得体育馆能尽可能满足多位客户的需求,并算出针对上表的10个客户申请,最多可以安排几位客户申请。



(1)该问题可以使用(\_\_\_)算法求解?

(2)该问题在计算前,按照(\_\_\_)从小到大排序进行预处理。

(3) 最多安排的客户编号顺序为(\_\_\_)、(\_\_\_)、(\_\_\_)、(\_\_\_).

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

贪心

**第二空：**

f(i)

**第三空：**

3

**第四空：**

6

**第五空：**

9

**第六空：**

7

窗体底端

17

有4种不同面值的硬币(每种无限多)。面值分别为 V1=3, V2=5, V3=7, V4=14。给定S=100,可以选用多少个硬币,使得面值之和恰好为S?输出硬币数目的最小值及组合。

(1)该问题最好使用(\_\_\_)算法求解?

(2) 输出硬币数目的最小值是(\_\_\_).

(3)输出硬币数目的最小值的组合是第1种(\_\_\_)枚、第2种(\_\_\_)枚、,第3种(\_\_\_)枚、第4种(\_\_\_)枚.

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

动态规划

**第二空：**

10

**第三空：**

2

**第四空：**

2

**第五空：**

0

**第六空：**

6

窗体底端

18

任务安排问题:某公司有5个工作岗位,每个岗位需要1个人,现接到5位待业者申请。A申请岗位1、2、3,B申请1、4,C和D申请4、5,E申请5。

最多\_\_\_人就业,相应安排是:A安排岗位\_\_\_\_或\_\_\_\_,B安排岗位\_\_\_ 。

如果n个人m个岗位,最好使用\_\_\_\_算法。

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

4

**第二空：**

2

**第三空：**

3

**第四空：**

1

**第五空：**

匈牙利算法

窗体底端

19

插入排序最好情况下的时间复杂度为O(\_\_\_),最坏情况下的复杂度为O（\_\_\_）

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

n

**第二空：**

n2

窗体底端

20

常见的两种分支限界法为      分支限界法和     分支限界法。

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

队列式

**第二空：**

优先队列式

窗体底端

21

QQ截图20190624101802.png

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

logn

窗体底端

22

现实计算机上的随机算法中使用的随机数都是\_\_\_\_。

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

伪随机数

窗体底端

23

贪心算法一般\_\_\_\_后再进行最优化选择。

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

预处理

窗体底端

24

快速排序的时间复杂度是O（  ）

窗体顶端

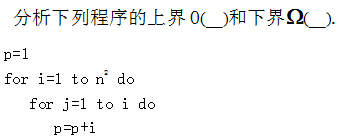
[正确答案](javascript:void(0))

**第一空：**

n2

窗体底端

25



窗体顶端

[正确答案](javascript:void(0))

**第一空：**

n^4

**第二空：**

n^4

窗体底端

26

设G = <V, E>中无孤立点，|V|=n，边覆盖数 + 匹配数 = \_\_\_

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

N

窗体底端

27

动态规划方程M[i，j]= min(M[i-1,j] + M[i-1,j-1] +wij), 1≤i≤k≤j≤n, 则算法的则算法的时间复杂度为O（\_\_\_\_）.

窗体顶端

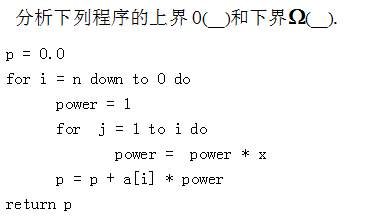
[正确答案](javascript:void(0))

**第一空：**

n2

窗体底端

28



窗体顶端

[正确答案](javascript:void(0))

**第一空：**

n^2

**第二空：**

n^2

窗体底端

29

逆删除算法利用了使用\_\_\_\_的特性

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

环

窗体底端

30

备忘录方法是\_\_\_\_算法的变形。

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

动态规划

窗体底端

31

无向图G = (V, E).M  E ，如果任意一个顶点至多出现在M中的一条边中，M是一个\_\_\_。

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

匹配

窗体底端

32

由结果倒过来推解前提条件，需要使用（   ）算法。

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

倒推

窗体底端

33

回溯法中，如果解空间树是排列树，所给的问题规模为n时，通常有\_\_个叶结点，遍历子集树需 O(   ) 计算时间 。

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

n!

**第二空：**

n!

窗体底端

34

冒泡排序的时间复杂度是O（\_\_\_\_）

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

n2

窗体底端

35

木板问题:农夫约翰为了修理栅栏,将一块木板切割成N块,N块的长度和=原木板长度。每次切割木板时的开销为该木板的长度。木板长15,切成长为1、 2 、3 、 4 、5的木板。该问题最好使用(\_\_)算法求解,切割的最小开销是(\_\_) ,第一次切割成长度为(\_\_\_和(\_\_)的两块。

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

贪心

**第二空：**

33

**第三空：**

6

**第四空：**

9

窗体底端

36

给定一个数字三角形,从顶至底有多条路径,每一步可沿左斜线向下或沿右斜线向下,路径所经过的数字之和为路径得分。最小路径得分为(\_\_),相应路径(从顶到底)对应的数字为(\_\_),(\_\_)(\_\_),(\_\_),(\_\_)。

        7

      3  8

    8  4  0

  2  7  4  4

4  5  2  6  5

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

20

**第二空：**

7

**第三空：**

3

**第四空：**

4

**第五空：**

4

**第六空：**

2

窗体底端

37

把任意一个解逐渐变为贪心算法的解，不会影响其最优性。这种证明方法是\_\_\_\_.

窗体顶端

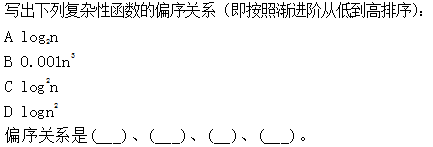
[正确答案](javascript:void(0))

**第一空：**

交换论证

窗体底端

38



窗体顶端

[正确答案](javascript:void(0))

**第一空：**

A

**第二空：**

D

**第三空：**

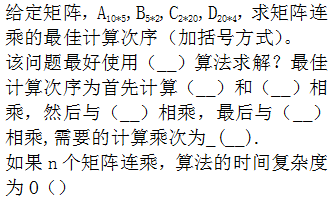
C

**第四空：**

B

窗体底端

39



窗体顶端

[正确答案](javascript:void(0))

**第一空：**

动态规划

**第二空：**

C

**第三空：**

D

**第四空：**

B

**第五空：**

A

**第六空：**

400

**第七空：**

n^3

窗体底端

40

选择排序的时间复杂度是O（\_\_\_\_）

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

n2

窗体底端

41

回答是与否的问题是\_\_\_问题。

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

判定

窗体底端

42

分析下列方程的上界O(\_\_)和下界Ω(\_\_).

T(n) = 3T(n/2) + n, T(1)=1

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

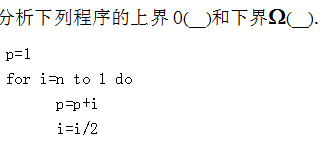
n^log3

**第二空：**

n^log3

窗体底端

43



窗体顶端

[正确答案](javascript:void(0))

**第一空：**

logn

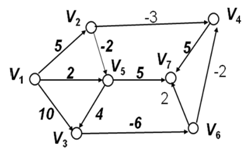
**第二空：**

logn

窗体底端

44

给定图,求V1到V7的最短路。



(1)该问题最好使用()算法求解?

(2) V1到V7的最短路为V1(\_\_)(\_\_)(\_\_)V7。

(3) V1到V7的最短路长度为\_\_\_\_。

(4) 如果求任意两点间最短路,一般使用(\_\_)算法求解。

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

动态规划

**第二空：**

V5

**第三空：**

V3

**第四空：**

V6

**第五空：**

2

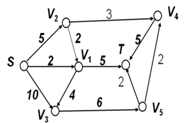
**第六空：**

Floyd;弗洛伊德

窗体底端

45

给定如下网络G,求最大流



(1) 最大网络流值是\_\_\_

(2) 最小割的容量是\_\_\_\_\_.

(3) 最小割包含T和顶点\_\_

(4) 如果G中有n个顶点m条边,最好使用\_\_\_\_算法。

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

11

**第二空：**

11

**第三空：**

V1

**第四空：**

网络流

窗体底端

46

回溯法的两种解空间树为     、     。

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

排列树

**第二空：**

子集树

窗体底端

47

任务安排问题:某公司有5个工作岗位,每个岗位需要1个人,现接到5位待业者申请。A申请岗位1、2、3,B申请1、4,C和D申请4、5,E申请5。

最多\_\_\_人就业,相应安排是:A安排岗位\_\_\_\_或\_\_\_\_,B安排岗位\_\_\_ 。

如果n个人m个岗位,最好使用\_\_\_\_算法。

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

4

**第二空：**

2

**第三空：**

3

**第四空：**

1

**第五空：**

匈牙利算法

窗体底端

48

QQ截图20190624093448.png

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

2^(n-1)

窗体底端

49

折半插入排序的时间复杂度是θ（\_\_\_\_）。

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

 n2

窗体底端

50

找n个元素的中位数的分治算法的时间复杂度为O(\_\_\_).

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

n

窗体底端

51

队列式分支限界法使用     的搜索方式。

窗体顶端

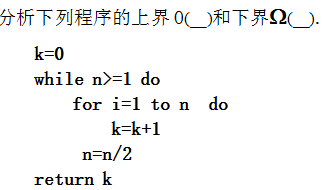
[正确答案](javascript:void(0))

**第一空：**

广度优先

窗体底端

52



窗体顶端

[正确答案](javascript:void(0))

**第一空：**

n

**第二空：**

n

窗体底端

53

如果对于同一实例，蒙特卡罗算法不会给出2个不同的正确解答，则称该蒙特卡罗算法是\_\_\_\_的。

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

一致

窗体底端

54

给定两张喜欢列表,喜欢程度从高到低. 一个婚姻介绍所开一个聚会,有n个男士,m个女士,每个男士i对于想约会的女士有一个优先列表Mi,每个女士j对于想约会的男士有一个优先列表Fj。设计一个约会算法,使男士和女士都比较满意。

男孩的喜欢列表M

|  |  |  |  |
| --- | --- | --- | --- |
| X | B | A | C |
| Y | B | C | A |
| Z | C | B | A |

女孩的喜欢列表F

|  |  |  |  |
| --- | --- | --- | --- |
| A | Y | Z | X |
| B | Z | X | Y |
| C | Y | Z | X |

(1)该问题可以使用(\_\_)算法求解

(2)示例得到的稳定匹配是:X和(\_\_),Y和(\_\_),Z和(\_\_).

(3)算法的时间复杂度为O(\_\_).

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

稳定匹配

**第二空：**

A

**第三空：**

C

**第四空：**

B

**第五空：**

nm;mn

窗体底端

55

20190603170508.png

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

n+m;m+n

窗体底端

56

QQ截图20190624093407.png

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

n^log3

**第二空：**

n^log3

窗体底端

57

多选式时间可验证问题是\_\_\_\_问题。

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

NP

窗体底端

58

回溯法解题步骤：

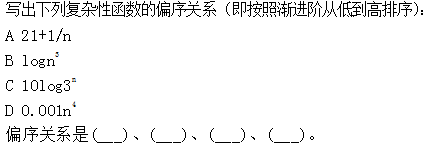
窗体顶端

[正确答案](javascript:void(0))

**第一空：**装载问题 **第二空：**旅行商问题

窗体底端

59



窗体顶端

[正确答案](javascript:void(0))

**第一空：**

A

**第二空：**

B

**第三空：**

C

**第四空：**

D

窗体底端

60

背包问题,背包容量C=20 ,物品价值p =[4, 8,15, 1, 6,3], 物品重量w=[5, 3,2, 10, 4, 8].如果是0-1背包问题,求装入背包的最大价值和相应装入物品。该问题最好使用(\_\_\_)算法求解. 装入背包的最大价值是(\_\_\_\_\_),对应的完整物品是(\_\_\_\_)、(\_\_\_\_)、(\_\_\_\_)、(\_\_\_)。

如果物品数为n,算法的时间复杂度为O()。

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

动态规划

**第二空：**

33

**第三空：**

1

**第四空：**

2

**第五空：**

3

**第六空：**

5

**第七空：**

nC;Cn

窗体底端

61

20190603170447.png

窗体顶端

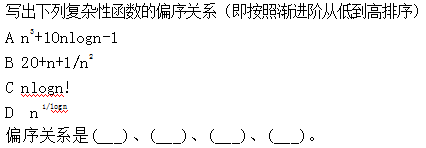
[正确答案](javascript:void(0))

**第一空：**

100

窗体底端

62



窗体顶端

[正确答案](javascript:void(0))

**第一空：**

D

**第二空：**

B

**第三空：**

C

**第四空：**

A

窗体底端

63

T(n) = 2T(n-1) , T(1)=1,则 T(n) =Ω(\_\_\_)

窗体顶端

[正确答案](javascript:void(0))

**第一空：**

2^(n-1)

窗体底端

64

P \_\_\_ NP

窗体顶端

[正确答案](javascript:void(0))

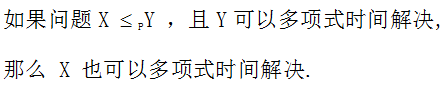
**第一空：**

包含于的符号

窗体底端

**四、判断题 （题数：210，考试抽取20道，分值20分）**

1

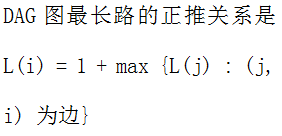


窗体顶端

正确答案： √

窗体底端

2



窗体顶端

正确答案： √

窗体底端

3

20190603170151.png

窗体顶端

正确答案： √

窗体底端

4

f(n)=O(g(n)) 则 2^f(n)=O(2^g(n))

窗体顶端

正确答案： √

窗体底端

5

现实计算机上无法产生真正的随机数。

窗体顶端

正确答案： √

窗体底端

6

分支限界法一般找出解空间树中满足约束条件的所有解。

窗体顶端

正确答案： ×

窗体底端

7

枚举法适用于问题的小规模实例。

窗体顶端

正确答案： √

窗体底端

8

位向量法生成子集，子集或运算可以生成差集

窗体顶端

正确答案： ×

窗体底端

9

增加拉斯维加斯算法的反复求解次数, 可使求解无效的概率任意小。

窗体顶端

正确答案： √

窗体底端

10

设G = <V, E>中无孤立点,|V|=n,则边覆盖数 + 匹配数 = n

窗体顶端

正确答案： √

窗体底端

11

任何排序算法至少需要O(n log n)次比较。

窗体顶端

正确答案： ×

窗体底端

12

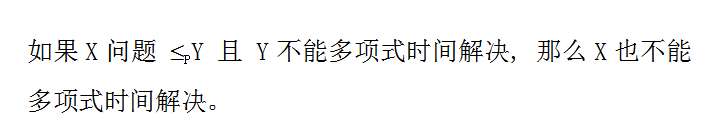
DAG动态规划算法中反推的开始点是无出边的顶点。

窗体顶端

正确答案： √

窗体底端

13



窗体顶端

正确答案： ×

窗体底端

14

20190603170211.png

窗体顶端

正确答案： √

窗体底端

15

贪心算法总能找到最优解

窗体顶端

正确答案： ×

窗体底端

16

算法必须在有穷时间终止。

窗体顶端

正确答案： √

窗体底端

17

舍伍德算法总是有解, 且解总是正确的,但平均性能未改变。

窗体顶端

正确答案： √

窗体底端

18

最短增广路算法可以设计出在O（logn）的平均时间内找到一条最短增广路，算法复杂度为O（mnlogn）

窗体顶端

正确答案： √

窗体底端

19

如果一个算法是多项式时间算法，该算法是有效的,是好算法。（

窗体顶端

正确答案： √

窗体底端

20

如果图G中每条边的权重都是互不相同的，图G必定只有一颗最小生成树。

窗体顶端

正确答案： √

窗体底端

21

20190603170133.png

窗体顶端

正确答案： ×

窗体底端

22

贪心选择的性质是贪心算法与动态规划算法的主要区别。

窗体顶端

正确答案： √

窗体底端

23

算法是一步步正确解决问题的方法和策略。

窗体顶端

正确答案： √

窗体底端

24

确定性算法的每一计算步骤都确定,求解同一实例用同一算法求解两次，所得结果完全相同。

窗体顶端

正确答案： √

窗体底端

25

程序必须在有穷时间终止。

窗体顶端

正确答案： ×

借助随机预处理技术，不改变原有的确定性算法，仅对其输入进行随机洗牌，可收到舍伍德算法的效果。

窗体顶端

正确答案： √

窗体底端

27

优先队列式分支限界法按照优先队列中规定的优先级，选取优先级最高的结点，成为当前扩展结点。

窗体顶端

正确答案： √

窗体底端

28

一般来说，递归的效率高于递推,

窗体顶端

正确答案： ×

窗体底端

29

问题的最优子结构性质是该问题可用贪心算法或动态规划算法求解的关键特征。

窗体顶端

正确答案： √

窗体底端

30

增量构造法生成子集,便于实现集合的操作。

窗体顶端

正确答案： ×

窗体底端

31

SPFA算法是Bellman-Ford算法的一种队列实现,适用于稀疏图。

窗体顶端

正确答案： √

窗体底端

32

图 G 是二分图当且仅当G中无奇数长的环.

窗体顶端

正确答案： √

窗体底端

33  
20190603170606.png

窗体顶端

正确答案： ×

窗体底端

34

证明算法不正确,需要证明对任意实例算法都不能正确处理。

窗体顶端

正确答案： ×

窗体底端

35

递推是从问题的最终目标出发，逐渐将复杂问题化为简单问题，最终求得问题。

窗体顶端

正确答案： ×

窗体底端

36

回溯法中，如果解空间树是子集树，当所给的问题规模为n时，通常有2n个叶结点，遍历子集树需O(2n)计算时间 。

窗体顶端

正确答案： √

窗体底端

37

20190603180159.png

窗体顶端

正确答案： ×

窗体底端

38

回溯法用限界函数剪去得不到最优解的子树

窗体顶端

正确答案： √

窗体底端

39重复

一般来说，递归的效率高于递推,

窗体顶端

正确答案： ×

窗体底端

40

DAG图最长路的递推函数d(i)表示从某个顶点i出发的最长路长度 。

窗体顶端

正确答案： √

窗体底端

41

死结点是正在产生儿子的结点

窗体顶端

正确答案： ×

窗体底端

42

算法可以使用自然语言描述,便于描述同一算法的多种含义。

窗体顶端

正确答案： ×

窗体底端

43

20190603170041.png

窗体顶端

正确答案： ×

窗体底端

44

算法的每一条指令必须要有确切的含义,必须是清楚的、无二义的。

窗体顶端

正确答案： √

窗体底端

45

增加蒙特卡罗算法的求解次数, 可使求解错误的概率任意小。

窗体顶端

正确答案： √

窗体底端

46

拉斯维加斯算法肯定得到一个正确解。（1.0分）

窗体顶端

正确答案： ×

窗体底端

47

20190603180422.png

窗体顶端

正确答案： √

窗体底端

48

对于给定的正整数n，判定n是一个素数的充要条件是(n-1)!≡1(mod n)。

窗体顶端

正确答案： √

窗体底端

49

网络流满足容量约束，但一般不满足流量守恒约束。

窗体顶端

正确答案： ×

窗体底端

50类似

递推是从问题的最终目标出发，逐渐将复杂问题化为简单问题，最终求得问题。

窗体顶端

正确答案： ×

窗体底端

51

QQ截图20190624104738.png

窗体顶端

正确答案： √

窗体底端

52

蒙特卡罗算法的结果肯定是一个正确解。

窗体顶端

正确答案： ×

窗体底端

53

QQ截图20190624103859.png

窗体顶端

正确答案： ×

窗体底端

54

分支限界算法的活结点一旦成为扩展结点,就一次性产生其所有儿子结点。

窗体顶端

正确答案： √

窗体底端

55

回溯算法中在一个扩展结点变成死结点之前,它一直是扩展结点。

窗体顶端

正确答案： ×

窗体底端

56

证明算法不正确，只需给出一个反例，算法不能正确处理即可。

窗体顶端

正确答案： √

窗体底端

57

一个问题的同一实例可以有不同的表示形式。

窗体顶端

正确答案： √

窗体底端

58

同一算法只有一种形式描述。

窗体顶端

正确答案： ×

窗体底端

59类似

确定型算法的每一计算步骤都确定,求解同一实例用同一算法求解两次,所得结果完全相同。

窗体顶端

正确答案： √

窗体底端

60

分支限界法找出满足约束条件的一个解，或是在满足约束条件的解中找出在某种意义下的最优解。

窗体顶端

正确答案： √

窗体底端

61

20190603170004.png

窗体顶端

正确答案： ×

窗体底端

62

20190603165914.png

窗体顶端

正确答案： √

窗体底端

63

分支限界法与回溯法都是在问题的解空间树T上搜索问题的解，二者搜索方式不同，但求解目标相同。

窗体顶端

正确答案： ×

窗体底端

64

DAG上最短路，固定起点和终点没有意义。

窗体顶端

正确答案： ×

窗体底端

65

20190603180242.png

窗体顶端

正确答案： √

窗体底端

66

哈夫曼编码的平均码长最小

窗体顶端

正确答案： √

窗体底端

67

把X的实例多项式时间变换为Y的输入,通过黑箱Y,得到X的解。则 X可多项式时间归约到Y

窗体顶端

正确答案： √

窗体底端

68

算法是解决特定问题的有穷指令序列。

窗体顶端

正确答案： √

窗体底端

69

使用限界函数作优先级, 第一个加入队列的叶子就是最优解

窗体顶端

正确答案： √

窗体底端

70

随机算法共同点是计算时间越多或运行次数越多,正确性越高.

窗体顶端

正确答案： √

窗体底端

71

有些问题采用倒推法，容易理解和解决。

窗体顶端

正确答案： √

窗体底端

72

设G是n阶无孤立点的图，则V\*是G的顶点覆盖，当且仅当V-V\*是G的独立集。

窗体顶端

正确答案： √

窗体底端

73

最小生成树是唯一的。

窗体顶端

正确答案： ×

窗体底端

74

计算机每次求解是针对问题的每个实例求解。

窗体顶端

正确答案： ×

窗体底端

75

20190603170651.png

窗体顶端

正确答案： ×

窗体底端

76

回溯法搜索解空间时，在搜索试探时选取x[i]的值顺序是任意的，顺序对于计算量没有差别。

窗体顶端

正确答案： ×

窗体底端

77

最大权独立集不包含u，可能包含儿子结点，也可能不包含儿子结点

窗体顶端

正确答案： √

窗体底端

78

Sherwood算法随机选择一个数组元素作为划分标准求解k小元素问题,保证线性时间的平均性能。

窗体顶端

正确答案： √

窗体底端

79

尾递归中的递归调用是整个函数体中最后执行的语句且它的返回值不属于表达式的一部分。

窗体顶端

正确答案： √

窗体底端

80

递推是从简单问题出发,一步步的向前发展,最终求得问题。

窗体顶端

正确答案： √ 我的答案：

窗体底端

81

拉斯维加斯算法肯定得到正确解或找不到解, 一旦找到一个解,一定是正确解

窗体顶端

正确答案： √

窗体底端

82

对于简单网络，最短增广路算法时间复杂度O(nm)

窗体顶端

正确答案： ×

窗体底端

83

同一数学模型使用不同的数据结构会有不同的算法，有效性有很大差别。

窗体顶端

正确答案： √

窗体底端

84

有多项式时间算法的问题是易解问题

窗体顶端

正确答案： √

窗体底端

85

最小堆中每个元素调整的次数不超过树高 Q(logn)。

窗体顶端

正确答案： √

窗体底端

86

分块查找一般设分块的长度是n/2.

窗体顶端

正确答案： ×

窗体底端

87

扩展结点是所有儿子已经产生的结点。

窗体顶端

正确答案： ×

窗体底端

88

蒙特卡罗算法的结果未必正确,并且可能难以有效判定是否正确。

窗体顶端

正确答案： √

窗体底端

89

分治算法的思想是将难以直接解决的大问题，分割成一些规模较小的子问题，以便各个击破，分而治之。

窗体顶端

正确答案： √

窗体底端

90

时间复杂度是指算法最好情况下的运行时间。

窗体顶端

正确答案： ×

窗体底端

91

对于稠密图,Floyd算法的效率要高于执行n次Dijkstra算法,也要高于执行n次SPFA算法

窗体顶端

正确答案： √

窗体底端

92

随机抽取数组元素k次,从最接近搜索元素x 的位置顺序搜索,顺序搜索的平均比较次数为O(n/(k+1)).

窗体顶端

正确答案： √

窗体底端

93

Kruskal算法的贪婪准则是每一次选取不构成环路的最小边。

窗体顶端

正确答案： √

窗体底端

94

回溯法搜索解空间时,在其它条件相当的前提下,让可取值最少的x[i]优先,可以减少计算。

窗体顶端

正确答案： √

窗体底端

95

优先队列式分支限界按照优先队列中规定的优先级,选取优先级最高的结点,成为当前扩展结点。

窗体顶端

正确答案： √

窗体底端

96

一个问题的算法必须在有穷时间终止，并且对一切合法的输入都能得出满足要求的结果。

窗体顶端

正确答案： √

窗体底端

97

f=ω(g)且g =ω(h)则f =ω(h)

窗体顶端

正确答案： √

窗体底端

98

分治法分解的子问题与原问题形式相同。

窗体顶端

正确答案： √

窗体底端

99重复

求解同一实例用同一随机化算法求解两次,所用时间和所得结果可能完全不同

窗体顶端

正确答案： √

窗体底端

100

设f任意流,(A, B)是任意 s-t割.则流值至多等于割的容量.

窗体顶端

正确答案： √

窗体底端

101

减治法减一个常量就是每次迭代减去一个相同的常数因子(一般为2)

窗体顶端

正确答案： ×

窗体底端

102

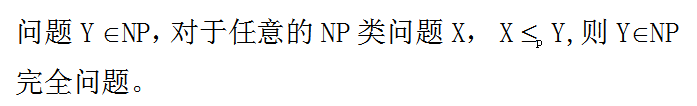
尾递归中的递归调用是整个函数体中最后执行的语句且它的返回值不属于表达式的一部分。

窗体顶端

正确答案： √

窗体底端

103



窗体顶端

正确答案： √

窗体底端

104

递归表现为自己调用自己，递推则没有这样的形式。

窗体顶端

正确答案： √

窗体底端

105

分治与递归都是从大规模问题逐步化为小规模问题，因此分治算法经常使用递归实现。

窗体顶端

正确答案： √

窗体底端

106

循环用于重复性的工作。循环体的特点是：“以不变应万变”。

窗体顶端

正确答案： √

窗体底端

107

贪心选择通过一步步选择得到问题的解，每一步的局部最优解都构成全局最优解的一部分。

窗体顶端

正确答案： √

窗体底端

108

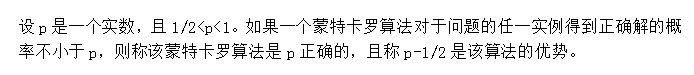
每个迭代算法原则上总可以转换成与它等价的递归算法;反之不然。

窗体顶端

正确答案： √

窗体底端

109



窗体顶端

正确答案： √

窗体底端

110

回溯法是一种选优搜索法，按选优条件向前搜索，以达到目标。但是，当探索到某一步时，发现原先选择并不优或达不到目标，就退回一步重新选择，这种走不通就退回再走的技术，称为回溯法。

窗体顶端

正确答案： √

窗体底端

111

如果e是图G中权重最小的边,它肯定是G的一颗最小生成树的边。

窗体顶端

正确答案： √

窗体底端

112

Floyd算法是动态规划算法,稠密图效果最佳,边权可正可负。

窗体顶端

正确答案： √

窗体底端

113

0/1背包问题的动态规划算法是多项式时间算法。

窗体顶端

正确答案： ×

窗体底端

114

三分法的判定树是三叉树

窗体顶端

正确答案： √

窗体底端

115

队列式分支限界法以最小耗费优先的方式搜索解空间树。

窗体顶端

正确答案： ×

窗体底端

116

二进制法生成子集，子集与运算可以生成并集

窗体顶端

正确答案： ×

窗体底端

117

存在割(A, B) 使流值v(f)=割的容量cap(A, B).，则割 (A, B)是最小割。

窗体顶端

正确答案： √

窗体底端

118

正推是从小规模的问题推解出大规模间题的一种方法

窗体顶端

正确答案： √

窗体底端

119

给定连通图G, BFS遍历得到层次图，如果同一层中的结点无边相连，则G是二分图。

窗体顶端

正确答案： √

窗体底端

120

20190603175347.png

窗体顶端

正确答案： √

窗体底端

121

Dijkstra算法在求解过程中，源点到集合S内各顶点的最短路径一旦求出，则之后不变了，修改的仅仅是源点到还没选择的顶点的最短路径长度。

窗体顶端

正确答案： √

窗体底端

122

减少枚举变量可以减少枚举算法的时间复杂度。

窗体顶端

正确答案： √

窗体底端

123

如果存在割 (A, B) 使流值 v(f) = 割的容量cap(A, B),f为最大流。

窗体顶端

正确答案： √

窗体底端

124

增量构造法生成子集前需要对集合中元素从小到大排列。

窗体顶端

正确答案： √

窗体底端

125重复

如果对于X的任意实例，通过多项式次的计算步骤，加多项式次调用Y的算法，可解决X，则 X可多项式时间归约到Y。

窗体顶端

正确答案： √

窗体底端

126

舍伍德算法总是有解, 且解总是正确的，但最坏性能未改变。

窗体顶端

正确答案： ×

窗体底端

127

回溯法和分支限界法的主要区别在于，回溯法求取问题的一个解或所有解。（1.0分）

**0.0** 分

窗体顶端

正确答案： √ 我的答案：

窗体底端

128

给定网络 N=(V, E)的一个流 f ,任意一个节点满足流出量等于流入量。

窗体顶端

正确答案： ×

窗体底端

129

递归算法是直接或间接地调用自身的算法。

窗体顶端

正确答案： √

窗体底端

130

如果一个NP完全问题能在多项式时间内得到解决，那么NP中的每一个问题都可以在多项式时间内求解。

窗体顶端

正确答案： √

窗体底端

131重复

递归算法是直接或间接地调用自身的算法。

窗体顶端

正确答案： √

窗体底端

132

好的约束函数能显著地减少所生成的结点数。但这样的约束函数往往计算量较大。因此，在选择约束函数时通常存在生成结点数与约束函数计算量之间的折衷。

窗体顶端

正确答案： √

窗体底端

133

回溯法的一个显著特征是在搜索过程中动态产生问题的解空间。

窗体顶端

正确答案： √

窗体底端

134

好算法具有如下特性:当输入规模加倍时,算法降低C倍,C为常数。

窗体顶端

正确答案： √

窗体底端

135

减治法减可变规模就是每次迭代规模减少的模式不同。

窗体顶端

正确答案： √

窗体底端

136

优先队列式分支限界法按照队列先进先出的原则，选取下一个节点为扩展结点。

窗体顶端

正确答案： ×

窗体底端

137

只有顶点i邻接的顶点j对i的影响独立时，才可用刷表法。

窗体顶端

正确答案： √

窗体底端

138

如果e是图G中权重最小的边，它至少是G的一颗最小生成树的边。

窗体顶端

正确答案： √

窗体底端

139

DAG动态规划算法中正推的开始点是无入边的顶点

窗体顶端

正确答案： √

窗体底端

140

回溯法不适用于解一些组合数相当大的问题。

窗体顶端

正确答案： ×

窗体底端

141

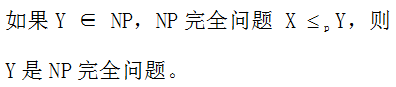
减少枚举变量的值域可以减少枚举算法的时间复杂度。

窗体顶端

正确答案： √

窗体底端

142重复



窗体顶端

正确答案： √

窗体底端

143重复

如果Y ∈ NP，NP完全问题 X pY，则Y是NP完全问题。

窗体顶端

正确答案： √

窗体底端

144重复

递归表现为自己调用自己，递推则没有这样的形式。

窗体顶端

正确答案： √

窗体底端

145

问题的两个要素是输入和实例。

窗体顶端

正确答案： ×

窗体底端

146

负权的单源最短路问题可以使用Dijkstra算法求解。

窗体顶端

正确答案： ×

窗体底端

147

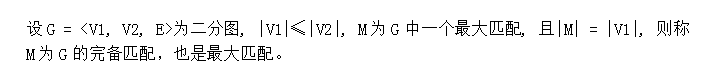
回溯法为了避免生成那些不可能产生最佳解的问题状态,不断地利用限界函数来处死那些实际上不可能产生所需解的活结点,以减少问题的计算量。

窗体顶端

正确答案： √

窗体底端

148



窗体顶端

正确答案： √

窗体底端

149

Bellman算法计算时,如果对于所有的v有OPT(n,v) = OPT(n-1,v) ,那么没有负环。

窗体顶端

正确答案： √

窗体底端

150

回溯法用约束函数在扩展结点处剪去不满足约束的子树。

窗体顶端

正确答案： √

窗体底端

151

不基于元素比较的排序算法可以在线性时间实现。

窗体顶端

正确答案： √

窗体底端

152

设G是n阶无孤立点的图,则V\*是G的最小顶点覆盖,当且仅当V-V\*是G的最大独立集。

窗体顶端

正确答案： √

窗体底端

153

递归方程是递归函数的要素之一 。

窗体顶端

正确答案： √

窗体底端

154

计算机每次求解只是针对一个实例求解,问题的描述针该问题的所有实例。

窗体顶端

正确答案： √

窗体底端

155

MST中若在树中任意增加一条边，将出现一个回路；若去掉一条边，将变成非连通图。

窗体顶端

正确答案： √

窗体底端

156

f=o(g)当且仅当g = ω (f)

窗体顶端

正确答案： √

窗体底端

157

队列式分支限界以最大效益优先方式产生状态空间树的结点。

窗体顶端

正确答案： ×

窗体底端

158重复

QQ截图20190624093836.png

窗体顶端

正确答案： √

窗体底端

159重复

分治法将原问题分解为若干个规模较小,相互独立,与原问题相似的子问题。

窗体顶端

正确答案： √

窗体底端

160

如果一个NP完全问题能在多项式时间内得到解决,那么P=NP。

窗体顶端

正确答案： √

窗体底端

161

算法是一个语句集合，按照顺序执行语句，处理实例，得到正确答案。

窗体顶端

正确答案： √

窗体底端

162

f=o(g) 且g = o(h)则f =o(h)

窗体顶端

正确答案： √

窗体底端

163

带需求的流通必须满足供给和 = 需求和.

窗体顶端

正确答案： √

窗体底端

164

匈牙利算法中起点和终点都是未匹配点的交错路径称为可增广路径，可增广路径有奇数条边。

窗体顶端

正确答案： √

窗体底端

165

重标号操作使它的标号上升到比周围最低的结点高度+1,使他的赢余能流出去

窗体顶端

正确答案： √

窗体底端

166

对于稀疏图，Floyd算法的效率要高于执行n次Dijkstra算法，也要高于执行n次SPFA算法

窗体顶端

正确答案： ×

窗体底端

167重复

正推是从小规模的问题推解出大规模间题的一种方法

窗体顶端

正确答案： √

窗体底端

168重复

使用限界函数作优先级, 第一个扩展的叶子就是最优解

窗体顶端

正确答案： √

窗体底端

169

http://p.ananas.chaoxing.com/star3/origin/cec0de1a465f2cbbbe1152a2220dbd3a.gif

窗体顶端

正确答案： ×

窗体底端

170

给定二分图G = <V, E>中无孤立点,|V|=n,其最大流算法求得最大流f, 则 G的最大独立数=n-f

窗体顶端

正确答案： √

窗体底端

171

分支限界法不能解决0/1背包问题

窗体顶端

正确答案： ×

窗体底端

172

20190603170626.png

窗体顶端

正确答案： ×

窗体底端

173

Floyd算法适用于多源最短路径,使用权矩阵代替邻接矩阵。

窗体顶端

正确答案： √

窗体底端

174

同一问题可能有几种不同的算法,解题思路和解题速度也会显著不同。

窗体顶端

正确答案： √

窗体底端

175

如果图G中每条边的权重都是互不相同的,图G可能存在多颗最小生成树。

窗体顶端

正确答案： ×

窗体底端

176

O(f)+O(g)=O(f+g),因此并行语句时间复杂度是两者的复杂度之和。

窗体顶端

正确答案： √

窗体底端

177

动态规划算法把原问题分为交叉的子问题，解决子问题，记录子问题的解，合并为原问题的解。

窗体顶端

正确答案： √

窗体底端

178

有下界的流通问题不一定有可行流。

窗体顶端

正确答案： √

窗体底端

179

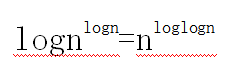
如果一个算法能应用于问题的任意实例,并保证得到正确解答,称这个算法解答了该问题。

窗体顶端

正确答案： √

窗体底端

180



窗体顶端

正确答案： √

窗体底端

181

对于问题的一个实例,解向量满足显式约束条件的所有多元组,构成了该实例的一个解空间。

窗体顶端

正确答案： √

窗体底端

182

设C是一个环， f 是C中的最大边，那么最小生成树中肯定包含f.

窗体顶端

正确答案： ×

窗体底端

183

问题A的实例可以变换为另一个问题B的实例。如果问题B的求解算法是已知的,那么问题A也可以求解。

窗体顶端

正确答案： √

窗体底端

184

回溯法是按广度优先策略搜索解空间树。

窗体顶端

正确答案： ×

窗体底端

185

给定一个实例，如果一个算法能得到正确解答，称这个算法解答了该问题

窗体顶端

正确答案： ×

窗体底端

186

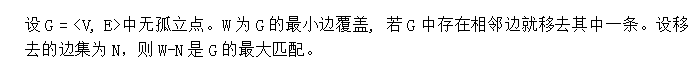
20190603180400.png

窗体顶端

正确答案： √

窗体底端

187



窗体顶端

正确答案： √

窗体底端

188

f=O(g) 且g = O(h)则f =O(h)

窗体顶端

正确答案： √

窗体底端

189

SPFA算法通过维护一个队列,使得一个节点的当前最短路径被更新之后没有必要立刻去更新其他的节点,大大减少了重复操作的次数。

窗体顶端

正确答案： √

窗体底端

190重复

时间复杂度是指算法最坏情况下的运行时间。

窗体顶端

正确答案： √

窗体底端

191

循环用于重复性的工作。循环体的特点是：“以不变应万变”。

窗体顶端

正确答案： √

窗体底端

192

增量构造法生成子集，不直接构造子集本身。

窗体顶端

正确答案： ×

窗体底端

193

回溯法在任何时刻,算法只保存从根结点到当前扩展结点的路径。

窗体顶端

正确答案： √

窗体底端

194

同一个问题可以有多种方式刻划它的最优子结构,求解速度、占用空间会有显著不同。

窗体顶端

正确答案： √

窗体底端

195

20190603175720.png

窗体顶端

正确答案： ×

窗体底端

196

QQ截图20190624104802.png

窗体顶端

正确答案： ×

窗体底端

197

EXP类是所有指数时间可解的判定问题组成的问题类

窗体顶端

正确答案： √

窗体底端

198

改进子问题合并的时间复杂度可以减少分治算法的时间。

窗体顶端

正确答案： √

窗体底端

199

减少枚举变量可以减少枚举算法的时间复杂度。

窗体顶端

正确答案： √

窗体底端

200

分支限界法在对问题的解空间树进行搜索的方法中，一个活结点有多次机会成为活结点。

窗体顶端

正确答案： ×

窗体底端

201

优先队列式分支限界法以最小耗费的方式产生状态空间树的结点。

窗体顶端

正确答案： √

窗体底端

202

分支限界法与回溯法，都是在问题的解空间树上搜索问题解。

窗体顶端

正确答案： √

窗体底端

203

活结点是自身已生成,但其儿子还没有全部生成的结点。

窗体顶端

正确答案： √

窗体底端

204

递归是从问题的最终目标出发,逐渐将复杂问题化为简单问题,最终求得问题

窗体顶端

正确答案： √

窗体底端

205重复

时间复杂度是指算法最坏情况下的运行时间。

窗体顶端

正确答案： √

窗体底端

206

子集生成算法中一般需要对集合元素进行定序。

窗体顶端

正确答案： √

窗体底端

207

贪心算法总能找到可行解，但未必是最优解

窗体顶端

正确答案： √

窗体底端

208

N个元素排序的时间复杂度不可能是线性时间。

窗体顶端

正确答案： ×

窗体底端

209

QQ截图20190624102441.png

窗体顶端

正确答案： ×

窗体底端

210

一般来说,迭代的效率高于递归。

窗体顶端

正确答案： √

窗体底端

**五、简答题 （题数：1，考试简答非此简答）**

1

队列式分支限界法使用     的搜索方式。

窗体顶端

[正确答案](javascript:void(0))

广度优先

窗体底端