

## 数据结构:

### 1.判断(10\*2'):

- 1)  $T(n)=a$ , 无论常数  $a$  多大, 时间复杂度为  $T(N)=T(n/2)+O(1)$  的解总是  $O(\log n)$
- 2) 基于 CBA 的算法对所有大小为  $n$  的数组时间复杂度是  $\Omega(n \log n)$
- 3) 基数排序的底层排序算法一定是稳定的
- 4) 输入随机的情况下完全二叉堆的插入平均时间是常数
- 5) 伸展树插入操作的分摊时间复杂度  $O(\log n)$
- 6) 对长度为  $m=4k+3$  素数的散列表双平方探测一定能访问其全部元素
- 7) 没改进的 next 算法时间复杂度也是  $O(n)$
- 8) Fib 查找时以前后黄金分割点作为轴点的常数相同
- 9) PFC(最优前缀编码)互换不同深度节点位置一定会破坏其性质
- 10) 任何情况下折半查找都比顺序查找快

### 2.选择(8\*3'):

- 1) 就地算法的空间复杂度是 A.  $O(1)$  B. C. D.
- 2) 后缀表达式扣去一个符号来猜扣去的是什么, 跟去年的类似
- 3) 一个非法表达式, 问强行求解的值是多少
- 4) 7 阶 B-树根节点常驻内存, 则对规模为 2017 的 B-树最多需要几次访问?  
A. B. C. D.
- 5) 散列长为 2017, 采用单平方探测, 已经存入 1000 个元素, 问此时最多有()个懒惰删除的桶单元  
A. 8 B. 9 C. D.
- 6) 分别按照递增和递减的顺序依次向平衡二叉树插入元素, 则存在常数  $k$  使  $n=2^k-1$  是二者生成的平衡二叉树相等的  
A. 充要条件 B. 必要不充分条件 C. 充分不必要条件 D. 不充分不必要条件
- 7) 左式堆最右侧链长度为  $k$ , 则左式堆\_\_含有\_\_个元素。  
A. 最少  $2^k$  B. 最少  $2^{k-1}$  C. 最多 \*\* D. 最多 \*\*
- 8)  $gs[0]=1$  的概率是  
A.  $1/m$  B.  $1/2^{(m-1)}$  C.  $1/2^m$  D.  $1/2^{(m+1)}$

### 3.单峰向量(13')

已知  $A[0, n)$ ,  $A[0 \sim k)$  严格单调递增,  $A[k \sim n)$  严格单调递减, 设计一个  $O(\log n)$  算法找出  $k$

- 1) 伪代码描述算法
- 2) 说明算法正确性
- 3) 证明最坏情况下时间复杂度也是  $O(\log n)$

### 4.最大和区间(13')

给定一个整数序列, 求出连续子序列和的最大值

- 1) 说明算法思路
- 2) 伪代码描述算法
- 3) 说明时间复杂度和空间复杂度

题注(大致意思): 蛮力算法就不要用啦, 是  $O(n^3)$ , 只有设计出  $O(n)$  算法才有可能满分,  $O(n^2)$  酌情给分。

## 计算机原理：

### 1.判断

- 1)提高 cpu 主频可以加快程序执行速度
- 2)raid6 坏两个磁盘也可以工作
- 3)c 语言若  $\text{int } x, y$  若  $x > y$ , 则  $-x < -y$

### 2.填空

- 1)-2017 的 32 位补码表示\_\_(16 进制或 2 进制)。
- 2)-2017 的 IEEE 单精度浮点表示\_\_。
- 3)高速缓存器的几种映射方式\_\_、\_\_、\_\_。
- 4)处理机\_\_逻辑电路进行算术运算，\_\_逻辑电路用于数据暂存，\_\_逻辑电路用于分支选择。

### 3.选择

- 1)以下关于五段流水线的处理机说法错误的是  
A.多个处理器不会发生结构冲突  
B.每个周期执行一个功能  
C.可以采用微程序或者硬连线设计  
D.不同的指令执行时间相同
- 2)以下说法正确的是  
A.缓存越大程序执行速度越快  
B.TLB 也是一种缓存数据和指令的缓存器  
C.  
D.
- 3)以下哪个不是响应异常的处理 A.保存 pc B.保存通用寄存器 C.保存异常原因 D.恢复 pc
- 4)以下哪种不可以解决数据冲突 A.暂停流水线 B.分支预测 C.调整指令顺序 D.数据旁路

4.五段流水线，每段 10ns，每个寄存器 5ns，以下一段程序(4 句)，问执行时间是多少

```
lw ***  
sub ***  
and ***  
or ***
```

## 计算机操作系统：

### 1.填空(10\*1')

- 1)父进程退出后，没结束的子进程变成\_\_。
- 2)高响应比调度算法的分子是\_\_，分母是\_\_。
- 3)优先级反置指的是\_\_抢占了\_\_的资源，\_\_时低优先级进程能动态改变优先级
- 4)\_\_支持暂时放弃互斥资源访问权，等待信号

### 2.判断

- 1)管程就是一个黑箱子，程序员往里面扔函数，同一时间只有一个函数在执行

- 2) Buddy 算法中，释放一个空间后可以根据起始长度和大小与相邻空闲空间合并
- 3) 如果用户强制使用任务管理器 kill 一个进程，那么即使它处于就绪状态/阻塞状态，操作系统也要把它变成运行状态
- 4) 操作系统采用 copy on write 机制时，fork() 函数会复制进程的页目录表
- 5) 管程和信号量在功能上等价

3. LRU、BEST、CLOCK、FIFO 页面置换算法是否能产生 belady 异常，若可以举出例子，不可以给出证明(6')

#### 4. ucore(6')

```

124=/* *
125 * struct Page - Page descriptor structures. Each Page describes one
126 * physical page. In kern/mm/pmm.h, you can find lots of useful functions
127 * that convert Page to other data types, such as physical address.
128 */
129=struct Page {
130     int ref; // page frame's reference counter
131     uint32_t flags; // array of flags that describe the status of the page frame
132     unsigned int property; // used in buddy system, stores the order (the X in 2^X) of the continuous memory block
133     int zone_num; // used in buddy system, the No. of zone which the page belongs to
134     list_entry_t page_link; // free list link
135     list_entry_t pra_page_link; // used for pra (page replace algorithm)
136     uintptr_t pra_vaddr; // used for pra (page replace algorithm)
137 };

// convert list entry to page
#define le2page(le, member) \
    to_struct((le), struct Page, member)

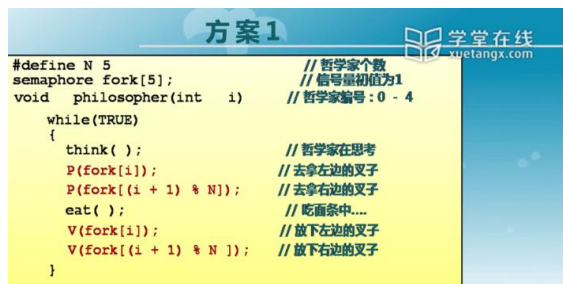
68
69=/* *
70 * to_struct - get the struct from a ptr
71 * @ptr: a struct pointer of member
72 * @type: the type of the struct this is embedded in
73 * @member: the name of the member within the struct
74 */
75=
76=
77=
78=
79=
80=
81=
82=
83=
84=
85=
86=
87=
88=
89=
90=
91=
92=
93=
94=
95=
96=
97=
98=
99=
100=
101=
102=
103=
104=
105=
106=
107=
108=
109=
110=
111=
112=
113=
114=
115=
116=
117=
118=
119=
120=
121=
122=
123=
124=
125=
126=
127=
128=
129=
130=
131=
132=
133=
134=
135=
136=
137=
138=
139=
140=
141=
142=
143=
144=
145=
146=
147=
148=
149=
150=
151=
152=
153=
154=
155=
156=
157=
158=
159=
160=
161=
162=
163=
164=
165=
166=
167=
168=
169=
170=
171=
172=
173=
174=
175=
176=
177=
178=
179=
180=
181=
182=
183=
184=
185=
186=
187=
188=
189=
190=
191=
192=
193=
194=
195=
196=
197=
198=
199=
200=
201=
202=
203=
204=
205=
206=
207=
208=
209=
210=
211=
212=
213=
214=
215=
216=
217=
218=
219=
220=
221=
222=
223=
224=
225=
226=
227=
228=
229=
230=
231=
232=
233=
234=
235=
236=
237=
238=
239=
240=
241=
242=
243=
244=
245=
246=
247=
248=
249=
250=
251=
252=
253=
254=
255=
256=
257=
258=
259=
260=
261=
262=
263=
264=
265=
266=
267=
268=
269=
270=
271=
272=
273=
274=
275=
276=
277=
278=
279=
280=
281=
282=
283=
284=
285=
286=
287=
288=
289=
290=
291=
292=
293=
294=
295=
296=
297=
298=
299=
300=
301=
302=
303=
304=
305=
306=
307=
308=
309=
310=
311=
312=
313=
314=
315=
316=
317=
318=
319=
320=
321=
322=
323=
324=
325=
326=
327=
328=
329=
330=
331=
332=
333=
334=
335=
336=
337=
338=
339=
340=
341=
342=
343=
344=
345=
346=
347=
348=
349=
350=
351=
352=
353=
354=
355=
356=
357=
358=
359=
360=
361=
362=
363=
364=
365=
366=
367=
368=
369=
370=
371=
372=
373=
374=
375=
376=
377=
378=
379=
380=
381=
382=
383=
384=
385=
386=
387=
388=
389=
390=
391=
392=
393=
394=
395=
396=
397=
398=
399=
400=
401=
402=
403=
404=
405=
406=
407=
408=
409=
410=
411=
412=
413=
414=
415=
416=
417=
418=
419=
420=
421=
422=
423=
424=
425=
426=
427=
428=
429=
430=
431=
432=
433=
434=
435=
436=
437=
438=
439=
440=
441=
442=
443=
444=
445=
446=
447=
448=
449=
450=
451=
452=
453=
454=
455=
456=
457=
458=
459=
460=
461=
462=
463=
464=
465=
466=
467=
468=
469=
470=
471=
472=
473=
474=
475=
476=
477=
478=
479=
480=
481=
482=
483=
484=
485=
486=
487=
488=
489=
490=
491=
492=
493=
494=
495=
496=
497=
498=
499=
500=
501=
502=
503=
504=
505=
506=
507=
508=
509=
510=
511=
512=
513=
514=
515=
516=
517=
518=
519=
520=
521=
522=
523=
524=
525=
526=
527=
528=
529=
530=
531=
532=
533=
534=
535=
536=
537=
538=
539=
540=
541=
542=
543=
544=
545=
546=
547=
548=
549=
550=
551=
552=
553=
554=
555=
556=
557=
558=
559=
560=
561=
562=
563=
564=
565=
566=
567=
568=
569=
570=
571=
572=
573=
574=
575=
576=
577=
578=
579=
580=
581=
582=
583=
584=
585=
586=
587=
588=
589=
590=
591=
592=
593=
594=
595=
596=
597=
598=
599=
600=
601=
602=
603=
604=
605=
606=
607=
608=
609=
610=
611=
612=
613=
614=
615=
616=
617=
618=
619=
620=
621=
622=
623=
624=
625=
626=
627=
628=
629=
630=
631=
632=
633=
634=
635=
636=
637=
638=
639=
640=
641=
642=
643=
644=
645=
646=
647=
648=
649=
650=
651=
652=
653=
654=
655=
656=
657=
658=
659=
660=
661=
662=
663=
664=
665=
666=
667=
668=
669=
670=
671=
672=
673=
674=
675=
676=
677=
678=
679=
680=
681=
682=
683=
684=
685=
686=
687=
688=
689=
690=
691=
692=
693=
694=
695=
696=
697=
698=
699=
700=
701=
702=
703=
704=
705=
706=
707=
708=
709=
710=
711=
712=
713=
714=
715=
716=
717=
718=
719=
720=
721=
722=
723=
724=
725=
726=
727=
728=
729=
730=
731=
732=
733=
734=
735=
736=
737=
738=
739=
740=
741=
742=
743=
744=
745=
746=
747=
748=
749=
750=
751=
752=
753=
754=
755=
756=
757=
758=
759=
760=
761=
762=
763=
764=
765=
766=
767=
768=
769=
770=
771=
772=
773=
774=
775=
776=
777=
778=
779=
780=
781=
782=
783=
784=
785=
786=
787=
788=
789=
790=
791=
792=
793=
794=
795=
796=
797=
798=
799=
800=
801=
802=
803=
804=
805=
806=
807=
808=
809=
810=
811=
812=
813=
814=
815=
816=
817=
818=
819=
820=
821=
822=
823=
824=
825=
826=
827=
828=
829=
830=
831=
832=
833=
834=
835=
836=
837=
838=
839=
840=
841=
842=
843=
844=
845=
846=
847=
848=
849=
850=
851=
852=
853=
854=
855=
856=
857=
858=
859=
860=
861=
862=
863=
864=
865=
866=
867=
868=
869=
870=
871=
872=
873=
874=
875=
876=
877=
878=
879=
880=
881=
882=
883=
884=
885=
886=
887=
888=
889=
890=
891=
892=
893=
894=
895=
896=
897=
898=
899=
900=
901=
902=
903=
904=
905=
906=
907=
908=
909=
910=
911=
912=
913=
914=
915=
916=
917=
918=
919=
920=
921=
922=
923=
924=
925=
926=
927=
928=
929=
930=
931=
932=
933=
934=
935=
936=
937=
938=
939=
940=
941=
942=
943=
944=
945=
946=
947=
948=
949=
950=
951=
952=
953=
954=
955=
956=
957=
958=
959=
960=
961=
962=
963=
964=
965=
966=
967=
968=
969=
970=
971=
972=
973=
974=
975=
976=
977=
978=
979=
980=
981=
982=
983=
984=
985=
986=
987=
988=
989=
990=
991=
992=
993=
994=
995=
996=
997=
998=
999=
1000=

```

le2page(\*page,page\_link)语句都需要展开那些宏定义？说明这个语句的含义。(还有一段 ucore 代码是 buddy 算法的页面分配函数，好像跟这道题关系不大，就不贴了主要是没找到。)

5. 页面 4kB，页表项 32bit，最大能支持 4GB 的内存空间，现在有一种新技术能支持 64GB 空间，这时页表项变成 64bit，重新设计页表结构

#### 6. 哲学家就餐问题



## 计算机网络：

### 1.选择

1)TCP/IP 与 OSI

A.

B.OSI 从上到下依次是应用层，会话层，表示层，网际层，网络层，数据链路层，物理层

C.TCP/IP 从上到下依次是应用层，网络层，数据链路层，物理层

D.TCP/IP 适用场合比 OSI 更广

2)奈奎斯特定理适用于以下哪些场合

i.同轴电缆 ii.光纤 iii.红外线

A. B. C. D. (以上三种介质的排列组合)

3)两地相距 3000 公里(传播速度 6ms/公里) 最大帧 64 字节,采用 GBN 协议,带宽为 1.544Mbps, 则若要最大限度发挥网络带宽, 至少需要多少比特的序号

A. B. C. D.

4)选择重传协议, 序号为 0-7, 发送窗口为 7, 当数据发送不产生冲突是, 接收窗口最大值为多少

A.4 B.5 C.7 D.8

5)dns 相关问题

A.天猫双 12 购物, 不同地方两个人访问淘宝得到的 ip 一定相同

B.

C.存储 ip 是五元组

D.数据库集中存储

2.一道透明网桥的大题, 两个网桥三段子网, 建立转发表, 要求填表。

3.一道路由器大题, 两个路由器, 三段网络的最大帧长度分别为 1024,512,912, 报头长度分别为 14,12,12(数据不一定准确)。

拓扑结构: A—R1—R2—B (R1、R2 的 e0 接口分别连 A、B, e1 接口互连),

1) 分配给这个网络一个 192.166.1.0/24 的 ip, 划分子网, 使 A, B 子网中主机数量尽可能多, 写出子网以及 R1, R2 的 e0、e1 接口的 ip 地址和子网掩码

2) 现在要发送一个长度为 900B 的 tcp 数据段, tcp 首部 20B, ip 首部 20B, identification 值为 X, 问这个数据段经过 A-R1、R1-R2、R2-B 三段子网的时候 total\_length、identification、DF、MF、offset 值分别为多少(单词记得不准)

3) A 要给 B 发 7 个数据段, 建立了一个 TCP 连接, 已知往返时间为 RTT, 问从建立连接开

始到发送结束共持续了多长时间