2020年12月20日 星期日

刷題 317

2019

- 50158 注意把六個情況都要確實帶過，不要用想的可能會有錯
- 50162 注意如果把++括號的話，他會先做++再做括號外的事
  - Ex:(a++)%5000 先a=a+1 再除5000
- 51070 第二小題的邊界情況很多，要注意！
- 51073 要試著不要寫的那麼複雜！
- 50180 注意要寫(int 64)a*(int 64)b ，不是(int 64)(a*b);
- 50181寫的太複雜，可以有更直覺的方法
  - 我目前作法是一層一層，但是寫的時候根本不清楚自己在幹嘛！
  - 寫pseudo code前應該要知道自己要寫出什麼東西，而不是邊寫邊想～
- 50182 注意切割的方法，事後切跟事前切都要注意！
- 50183 注意相同按壓次數時的比較，到底哪個比較好？
- 50184注意剪枝的設定會不會影響答案，也注意遞回的最佳解初始設定有沒有用好
- 對於有重複呼叫函式（recursion）的程式，一有true就要return ，不然可能會被後面的東西洗掉（51089 51087)
- 50189 有不知名錯誤 要重寫，不要開int **pointer這種東西，很難維護

Ex:
```
Bool fun(big){
Bool t=false;
for(int 0~n){
   t=fun(small);
   if(t){return true;}
}
Return t;
}
```

- 51090 test 的array記憶體分配在外面，不然每次都開新的 bsearch功用不熟！

Pointer(感覺206可做)

#沒辦法知道pointer array的大小，要事先存取值

S#72 pointer 指位置（array）->array 也可以反過來找ptr

對於&,ptr還沒有很熟，還要再看一下

S#190 pointer 指向雙層array

a_1={1,2,…} a_2={3.4}

*ptr[]={a,b}
拿到a_i的第j個=>*(*ptr+i)+j)

- S100 const int 不能改變值，但可以改變位置，所以開一個一維的pointer來做sort
  - 注意不要用*ptr=*ptr2來做，因為目的是要取位置，但這樣只有取值
  - 二刷想法：注意要用const 來接 const
    - 不能 int a=(const int)b;

2017(完成)

- 50085 注意行跟列在array裡是相反的，造成超多混淆。
- 50086 遇到那種要開很大的陣列要開在main以外，不然會RE
- 50092 記得使用memset, isdigit 很有用
- 50093 注意delimiter 選取 且注意strstr用法（很好用 ）
- 50095注意要開大的array一定要再main function 外！
  - 注意另開一個ptr 做temp 的switch 不會真的換到
  - 開2 level array 可以用stage1[l1][w1]來代替stage[l1]+w1
- 50096注意store at 64-bit是什麼概念！重寫！
  - “Store from MSB” 是個重要的概念！
- 50097 在進行>>時，可以考慮利用(unsigned long long)x>> 來移就不用在額外另一個都是0都數字把MSB的1給&掉
  - 注意memcpy,memmove的用法，非常好用！
- 50101 qsort可以用搬動pointer的方法來避免搬動structure，減少時間
- 50102 不熟，需重寫。是難題！
- 50103 注意fseek 用法！
  - 如果用SEEK_END 記得offset 要-1
  - 如果已經fread 完了，代表SEEK_CUR 已經+1了！
- 50105 可以用fun(array+d) 來代表給函數赴與array[d] 以後的整個array
- 50107 注意qsort 不要用return (a-b) 來做，因為可能是小數點...，一定要用return 1 ,0 的方法！
  - 如果是用 pointer 來做struct qsort 的話，記得是用Stud *x=*(Stud **)a; Stud *y=*(Stud **)b; x,y 代表的是pointer. (c=(*x) 則是struct)

- 50109 困難的做不出來，就做簡單的
- 50110要交換node->left 跟node ->right 不能用tmp 來做，因為這是複製位置，所以tmp=node->left沒有用。解決方法為開心的Node(malloc)來重做。
  - Global variable要在function裡重新再設定一次,不然會錯
- 50111 對於recursion，用int 來return 遠比用global 設 判斷成功條件還快

Recursion (finished:28 38 238 104 230 53,223,29,30,58)

- 能夠多設立一些終止遞迴的條件，有助於加速程式。ex:如果是算sum的程式，如果sum>wanted 就終止。
- 課本的河內塔。
- 38可以有更好的方式，注意recursion爆收不是唯一解
- 53寫的太複雜了，需要重做
- 223 的方法太慢，是有特定解法的

Character（完成）

- isspace() 測試是不是空白
- 220 切記 c=='\0' 不是EOF ，不要被騙了！

String(都是難題，可以三刷)

- S130 注意strspn是找出現的字，而不是同樣序列
- 260 要把char變成string，記得是要用一個char a[2] , a[0]=[你要的] ，a[1]='\0'
- S276
  - 小寫變大寫：toupper
  - 要比較字串跟已知字串：先寫好一個char[](char[last]='0')，且用strcmp
    - 可以直接用skip[4][128]={"of",...}
- S270不知道錯在哪
- S47 left mid right中間的那個如何更新是關鍵，因為可能跟left的尾一樣，也可能跟mid的頭一樣，因此分類到哪邊都錯。所以要都考慮，用insertleft(right)，這樣才能同時考慮到兩邊（一開始考慮到mid，後來考慮到left）

Bit(完成！)

- S222 注意如果>> or <<超過原本的位元數，情況不能預測。還有TLE不知道錯在哪（實測有錯）？搞定了！

File(完成)

- 136 注意 binary file (rb,wb)時不要用fprintf fscranf
- 225 可以拿來練習Bsearch,tree ，記得是讀一整個struct

Advanced(完成)

- 10010 Memset 才能把2d array所有值都設成0 or -1
  - memset(source, -1, sizeof(source[0][0]) *1001 * 1001);
  - 可以有奇怪的disjoint set使時間變很快
- 10029 對於判斷 double 是否為0，要用#define zero(x) ((x) < 1e-4 && (x) > -1e-4)，不然-0.0000判斷不出來
  - **結論：不要對float/double 用== or !=**
- 10032
  - **開任何一個記憶體都要先給初始值**
  - String cmp for pointer (不是用array形式的):

```
int cmp(const void *a,const void *b){
    char *c=*(char **)a;
    char *d=*(char **)b;
    return strcmp(c,d);
}
```

  - Array 形式還是用這種

```
int cmp(const void*a,const void *b){
    char *c=(char *)a;
    char *d=(char *)b;
    if(strcmp(c,d)>0){
        return 1;
    }
    else{
        return -1;
    }
}
```

  - 直接做一顆tree比較快

- 10039 對於整個string 做修改時，for迴圈用str[I]!='\0'，而非I<strlen(str)
  - 也可以用len=strlen(str), I<len
- 10030
  - 計算一個數字1的個數

```
int __builtin_popcount (unsigned int x)
int __builtin_popcountl (unsigned long)
int __builtin_popcountll (unsigned long long)
```

  - 如果要進位超過16位，要寫(1lu)<<31(對於 32bit), (1llu)<<63 (64bit)
  - 注意B07902112寫的BITCOUNT 方法，可以學習
- 10052 讀一整行 fgets，用'\n'判斷反而會WA?!
- 10118 free(ptr) 只能用在釋放ptr指向的記憶體，ptr本身不會被釋放
  - 也就是說free 只能用在指標上
  - 利用function 做 釋放記憶體不明智，因為recursion本身也需要記憶體
- 10017 可以做BStree的練習（一定要做！）
- 10031 奇怪的一題。long double可以到10^38次方？！
- 10049 round: 四捨五入整數。一般的浮點數會自動四捨五入
- 10069 malloc後要initialized bstnode, 但他有寫好了，所以用NEW即可
  - malloc的(int *)要寫，不然會被當(void *)
- 10078 char *str 要先用malloc來指定陣列大小。
  - fun(char *str) 是pass by reference? 這個不熟要弄懂
    - C 的pass by reference 是用傳pointer，不是用&
- 10043 不知道怎麼對的，但把func pointer=static func 就可以了！

Structure（完成）

- 44 不知道什麼是enum
- 251 struct 可以直接等於（比較）

2015(完成)

- 50000 要寫出不複雜的if-else程式，要有好的架構；如果太複雜，代表可能寫的辦法不夠好
  - 此題從啟動條件開始看，比較簡潔
- 簡單的if-else題都會debug很久，不能看測資來debug，不然考試沒用！
- 50008 直接用ptr[l][j][k]的方式比較簡單
  - **ptra=*ptrb要多注意
  - 記得他是在*ptr的地方存NULL，不是ptr=NULL
  - 指標還是沒有到很熟
- 50009寫程式不夠細心！boundary condition都沒有考慮！
- 50010 strcmp(dest,&s[x]) 可以決定從x開始複製
- 不要再從測資裡面debug!
- 50012 不知為何錯？
- 50017 寫的方法太複雜，根本不用去抓括號兩段
- 50018 return *char 直接return string 就好
  - Header file ifndef 一定要寫,且一定要寫有哪些函式！**不然在.c檔裡會出問題！！**
- 50024 toupper(char) 小寫變大寫
- 50025 特別的recursion法，要學會！
- 50026 檔案處理很不熟！
- 50032 recursion return 在哪裡設要搞清楚！在每個function裡都一格一格跑，會進入無窮迴圈！fun是要跑每一格取或不取，才能通過
- 50034 優秀的recursion法，要再複習！

Stdlib(完成)

- 85 可以利用讀char 來加速讀數字
- bsearch 是在已排序的陣列中找(找不到練習題目～)
- 253 printf(“%02d”)可以控制前面要輸出二位，不足的話前面補零
  - printf(“%2d”)是不足的話前面補空格

- 76 string 的 qsort 要看string是怎麼配置的（參考前面）

Data structure（剩269）

- 134 sort two dim array:如果不是指標(malloc)的(array[][])，如果是就要用前面的方法

```
int cmp(const void *a,const void *b){
    int *c=(int *)a;
    int *d=(int *)b;
    return c[1]-d[1];
}
```

  - 不要用下面的方法寫
  -
  - ，NULL的記憶體配置不一樣

```
for(int i=1;i<n;i++){
    cur=root;
    while(1){
        if(cur==NULL){
            cur=newnode(data[i][0]);
            break;
        }
        if(cur->val>data[i][0]){
            /*if(cur->left==NULL){
                cur->left=newnode(data[i][0]);
                break;
            }*/
            cur=cur->left;
        }
        else{
            /*if(cur->right==NULL){
                cur->right=newnode(data[i][0]);
                break;
            }*/
            cur=cur->right;
        }
    }
}
```

- 65 fgets 不熟->會錯可以用gets
  - atoi好用！
  - 讀整行還是用while(a!='\n')最保險
  - 注意判斷括號的i++地方->錯不只一次了！
  - 其實根本不用建立ｔｒｅｅ，寫程式前要想好！
- 87 pointer怎麼跑的可以用畫圖會更好！
  - 注意他是指記憶體位置，這樣觀念就很清楚！

- 269 不知名錯誤？

2016（完成）

- 50047 注意function 用main 裡的變數有pass by reference的效果
- 50049 pointer array直接當一般array 用，不用加*. *ptr是a[i]的概念
- 50051 注意 string qsort 方法
- 50052 變數名稱取長一點，不容易重複
  - 盡量寫註解(最好每個不簡單的都寫)，容易debug!
- 50054 注意hash table觀念！
  - 注意char *a (pointer)就是 string,strcmp 不用再加*
- 50058 做遞迴時傳進fun的陣列記得要重新開一個，不然會有問題！
  - 除非先做有 然後再去除（maintain fun） 就不會出問題
- 50068
  - 如果要建立存pointer 的array 可以用:
    - Node *a[1000]
    - Node **a=malloc(sizeof(Node)*1000);
  - 注意這題只用存parent就好，因為我們不需要找隔壁的child!
- 50066 注意linked list 如果要替換第一個記得要從原本的地方換，因爲後來要指只能從唯一一個地方(指標)開始
- 50067 讀好題目再寫，注意題目的意思！
- 50070 記得free 不然會MLE 但要確定是有被配置的才能被free,注意不能free array , 只能free pointer
- 50071 從command line 讀檔名
  - ./a.out p10160-1.dat
  - int main(int argc, char *argv[]){
    - FILE *file=fopen(argv,”rb”);…
- 50073 不知名錯誤

2018

- 50121 太煩雜且有重複性的題目，記得把每個變數該有的值，大小比較做註解。雖然很麻煩，但debug更累！
- 50123 注意C 的fun 傳變數會有pass by reference的效果，要特別小心！ 要記得複製一個新的來買保險
- 50124？？如果.c 檔要用main.c檔的變數，加上extern int 變數; 即可
- 50129 注意如果要算是array第幾格，是&A[i]-&A[0]即可，不用除以sizeof(int)
- 50130
  - 如果要在程式內指定字串內容的話，要用char *a的形式
    - Char *a=“hello”
  - 錯誤訊息記得要讀完！不然會影響到後面～
  - strstr(“big”,”small”)：確認有沒有一段字串
- 50131 strchr (“target”,char c):確認target裡有沒有字元c
- 50133 不用把每個片段都複製出來，有更快的做法
- 50134 bit operation 有更快的方法，不同大小的東西可以相加（假設要加int，ans<<=32, ans+=int，就不需要or）、or、and...
- 50139 遇到GPA換算，可以先開
  - char[10][3]={“A+”,”A”,”A-”…}
  - Int score[10]={4.3,4,3.7…}
  - 用strcmp來對應，比起switch簡潔方便許多
- 50140
  - 注意fread 讀整個檔不是用EOF 是!=0
  - 可以直接把int =char 不用擔心會出現問題
    - 不需要把char &1 !=0 來判斷出到底數值為何！
    - 也可以把fwrite (int ,sizeof(char)…) 就是讀int的最後8位！
- 50141
  - 是一題重要的觀念題，可以注意指標的觀念
    - 指標只是指位置！
      - Node *a,*b;

    - a=b 是指a指到b目前指的位置
    - b=b->next不影響a
  - Node *ans=NULL ,* tmp=ans 是很不對的！
    - 不要去指NULL，要等到ans第一次被賦值的時候再指
- 50143
  - 注意不要在開一個function後才指定child=NULL，要在一開始就指定他是NULL，不然就是用return Node的方式
    - 正確：
      - fun(Node ans…)
        - if(fun(ans->left)==0)
        - Ans->left=NULL
    - 錯誤：
      - fun(Node ans…)
        - ans=NULL
- 50146
  - 注意不要去設定NULL的值
    - Ex:tmp->next=NULL
    - 後來：tmp->next=a;
  - 注意linked list起點是設ans==NULL的情況，不一定第一條list不會是NULL
- 50147 不用去存x,y座標，只要存radius就好！

Function

- 50 題可以不用麻煩的方法寫！

Array

- 235 要處理很多奇怪的情況，但關鍵在於要減少大數字相乘，注意像最大公因數、left=0 的情況，對於處理分數的問題都要注意！