

# 高點名師

給你最佳  
最精準的詳解!

## 高普考解題 & 講座 地特命題趨勢

連續30年人氣爆棚，題點  
超過10,000名上榜生！



中會  
會計學(概要)  
鄭泓(鄭凱文)



財政學(概要)  
經濟學(概要)、公經  
張政(張家璋)



審計  
陳仁易



政治學  
初錫(蘇世岳)



行政法(概要)  
陳熙哲



土地法、土地登記  
土地經濟學  
曾榮耀



行政學(概要)  
公共政策  
高凱(高凱傑)



各國  
考銓(概要)  
何昀峯



心理學(概要)  
黃以迦

## 10/19起線上開講！

詳細講座資訊 ▶▶▶



行政廉政	
10/19(二) 18:30	【高普考行政學院】f 高凱：行政學/公共政策
10/20(三) 18:30	【高普考行政學院】f 何昀峯：各國人事/考銓制度
10/20(三) 18:30	【高點高上高普特考公職】f 陳熙哲：行政法
10/23(六) 18:00	【高點線上影音學習】y 初錫：政治學
10/23(六) 18:00	【高點線上影音學習】y 黃以迦：心理學

商科會科	
10/19(二) 18:30	【高點高上高普特考公職】f 張政：財政學/公共經濟學/經濟學
10/20(三) 18:30	【高點會人會語】f 鄭泓：會計學/中級會計學
10/23(六) 18:00	【高點線上影音學習】y 陳仁易：審計

地政	
10/20(三) 18:30	【高點來勝不動產專班】f 曾榮耀：土地法/土地登記/土地經濟學

【知識達數位科技股份有限公司附設臺北市私立高上文理短期補習班】  
 【高點數位科技股份有限公司附設私立高點文理短期補習班】  
 【高點數位科技股份有限公司附設新竹市私立高點建國文理短期補習班】  
 【高點數位科技股份有限公司附設臺中市私立高點文理短期補習班】  
 【高點數位科技股份有限公司附設嘉義市私立高點建國文理短期補習班】  
 【高點數位科技股份有限公司附設臺南市私立高點文理短期補習班】  
 【高點數位科技股份有限公司附設高雄市私立高點文理短期補習班】

另有 政大·淡江·三峽·羅東·逢甲·東海·中技·中科·彰化·雲科·中正

台北市開封街一段2號8樓  
 桃園市中壢區中山路100號14樓  
 新竹市東區民族路7號4樓  
 台中市東區大智路36號2樓  
 嘉義市垂楊路400號7樓  
 台南市中西區中山路147號3樓之1  
 高雄市新興區中山一路308號8樓

北市教四字第32151號  
 府教習字第0990091487號  
 府教社字第1020399275號  
 中市教終字第1090019268  
 府教社字第1011513214號  
 南市教社字第09912575780號  
 高市教四字第0980051133號



# 《資料結構》

- 一、A為(8×4)矩陣、B為(4×10)矩陣、C為(10×3)矩陣、D為(3×20)矩陣、E為(20×4)矩陣，  
 (一)請列出此5個矩陣相乘 $A \times B \times C \times D \times E$ 所有可能的乘法順序(請用括號表示乘法順序)。(5分)  
 (二)請使用Dynamic Programming(動態規劃)的技巧計算出此五個矩陣相乘 $A \times B \times C \times D \times E$ 的最佳乘法順序(請用括號表示乘法順序)，使得五個矩陣相乘所需要花費的乘法數量最少。(15分)  
 (三)請列出此五個矩陣相乘所需要花費的最少乘法數量。(5分)(注意：未說明Dynamic Programming的計算過程，不予計分。)

試題評析	矩陣連乘的最佳順序，屬於Dynamic Programming常見的範例，原理必須清楚了解，才能正確計算，唯計算過程稍嫌繁複，考驗耐性。
考點命中	《資料結構》，高點文化出版，王致強編著，頁1-33~1-34。

答：

- (一)  $n+1$ 個矩陣的連乘，共有 $\frac{1}{n+1} \binom{2n}{n}$ 種乘法順序。因為 $n+1=5$ 個矩陣，所以  $n=4$  共有 $\frac{1}{5} \binom{8}{4} = 14$ 種乘法順序，如下：

$((A \times B) \times C) \times D) \times E$   
 $((A \times B) \times (C \times D)) \times E$   
 $(A \times (B \times (C \times D))) \times E$   
 $((A \times (B \times C)) \times D) \times E$   
 $(A \times ((B \times C) \times D)) \times E$   
 $((A \times B) \times C) \times (D \times E)$   
 $(A \times (B \times C)) \times (D \times E)$   
 $(A \times B) \times ((C \times D) \times E)$   
 $(A \times B) \times (C \times (D \times E))$   
 $A \times (((B \times C) \times D) \times E)$   
 $A \times ((B \times C) \times (D \times E))$   
 $A \times ((B \times (C \times D)) \times E)$   
 $A \times (B \times ((C \times D) \times E))$   
 $A \times (B \times (C \times (D \times E)))$

- (二) 使用動態程式規劃演算法計算： $M_1 \times M_2 \times \dots \times M_n$
- 使用一個陣列 $size[1..n+1]$ 來記錄每個矩陣 $M_i$ 的列數，換言之，矩陣 $M_i$ 是一個大小為 $size[i] \times size[i+1]$ 的矩陣，其中 $1 \leq i \leq n$ 。
  - 用 $cost[i][j]$ 代表 $M_i \times \dots \times M_j$ 表示其所需的乘法次數。
  - 用 $middle[i][j]$ 來記錄最後一個矩陣乘法的位置，如果 $middle[i][j]=k$ 時，表示分別先算出 $(M_i \times \dots \times M_k)$ 與 $(M_{k+1} \times \dots \times M_j)$ ，然後再將兩者做矩陣相乘 $(M_i \times \dots \times M_k) \times (M_{k+1} \times \dots \times M_j)$ ，計算出 $M_i \times \dots \times M_j$ 。
  - Dynamic Programming的 Recursion Function  

$$cost[i][j] = \min_{i \leq k < j} \{cost[i][k] + cost[k+1][j] + size[i] \times size[k+1] \times size[j+1]\}$$

本題程式片段設計如下

```

for (i=1; i<=n; i++){
    cost[i][i]=0; middle[i][i]=0;
}
for (t=2; t<=n; t++) {
    for (i=1, j=i+t-1; j<=n; i++, j++) {
        cost[i][j]=∞;
        for (k=i; k<j; k++) {

```

【版權所有，重製必究！】

```

c=cost[i][k]+cost[k+1][j]+size[i]*size[k+1]*size[j+1];
if (c<cost[i][j]) {
    cost[i][j]=c;
    middle[i][j]=k;
}
}
}
}

```

cost矩陣如下，灰底處為每格計算的最佳選擇：

i \ j	1	2	3	4	5
1	0	8×4×10 =320	min{ 320+0+8×10×3, 0+120+8×4×3} =min{560, 216} =216	min{ 216+0+8×3×20, 320+600+8×10×20, 0+360+8×4×20} =min{696,2520,1000} =696	min{ 696+0+8×20×4, 216+240+8×3×4, 320+360+8×10×4, 0+408+8×4×4} =min{1336,552,1000,536} =536
2		0	4×10×3 =120	min{ 120+0+4×3×20, 0+600+4×10×20} =min{360, 1400} =360	min{ 360+0+4×20×4, 120+240+4×3×4, 0+360+4×10×4} =min{680,408,520} =408
3			0	10×3×20 =600	min{ 600+0+10×20×4, 0+240+10×3×4} =min{1400, 360} =360
4				0	3×20×4 =240
5					0

最佳乘法順：A×((B×C)×(D×E))

(三) 最佳解的乘法數量=536

二、假設收銀機內銅板的集合 $S=\{\$50, \$20, \$20, \$15, \$10, \$2, \$1, \$1, \$1\}$ ，而預計找錢給顧客的金額 $W=\$75$ 。(一)請設計一個Greedy (貪婪) 的演算法，來解決找錢給顧客的問題，使得找給顧客金額 $W$ 所使用的銅板數量最少，並依此Greedy的演算法列出找給顧客金額 $W=\$75$ 的過程。(15分)(二)此Greedy演算法適合使用何種資料結構來完成。(5分)(三)此Greedy演算法的解法是否能保證為最佳解？請舉例說明。(5分)

試題評析	本題屬於演算法問題，測驗考生對Greedy演算法特性、資料結構和Greedy演算法的弱點。零錢兌換款(Coin Change)問題，類似部份集合總和問題(Sum of Subset)，若要求解或是求最佳解，使用產生所有部份集合的遞迴程式，再加以修改以計算總和來解決問題。如果每種面額硬幣不限個數的情況，則一般使用Dynamic Programming來解決。
考點命中	王致強(2021)，《資料結構重點整理》，高點文化，頁5-19。

答：

(一)由大面額逐漸考慮到小面額，較可能可以換得較少個硬幣，演算法如下：

```

由大而小排序 S[1..n];
sum←0;
count←0;
for i←1 to n do {
    if (sum+S[i] ≤ W) {
        sum←sum+S[i];
        count←count+1;
    }
    if (sum==W) {
        output count;
        exit;
    }
}
output no solution;

```

計算過程如下表：

銅板 S[i]	\$50	\$20	\$20	\$15	\$10	\$2	\$1	\$1	\$1
累計金額 sum	\$50	\$70	\$70	\$70	\$70	\$72	\$73	\$74	\$75
硬幣數量 count	1	2	2	2	2	3	4	5	6

Greedy換得6個硬幣

(二)Greedy演算法適合使用2種方式來實作：

(1)排序法：視問題需求，進行遞增或遞減排序之後，以方便處理。

(2)Priority Queue：使用Heap 一類的結構，可以extract-max或extract-min最得最大或最小的項目進行Greedy處理。

(三)本題Greedy演算法未必可以找到最佳解，以題目整組資料的最佳解： $\$50+\$15+\$10=\$75$ ，最佳解只需要兌換3個硬幣，所以greedy演算法不見得能保證找到最佳解。

三、二元搜尋法 (binary search) 使用divide-and-conquer (分而治之) 演算法技巧，對一個已排序的 (sorted) 且長度為n的陣列A[0:n-1]，以二元化方式進行資料值x的搜尋，其最差時間複雜度 (worst case time complexity) 可降到 $\Theta(\log n)$ 。(一)請使用C++或Python語言，修改此二元搜尋法，使其能對未排序的 (unsorted) 且長度為n的陣列A[0:n-1]，進行三元化搜尋，即以divide-and-conquer技巧將此陣列切成三個子陣列，並在可能包含資料值x的子陣列繼續進行divide-and-conquer技巧的搜尋，如果找到則回傳1，如果找不到則回傳0。(17分) (注意：請寫一個searching 類別，內含一個search功能) (二)請分析修改後的三元化搜尋法其最差時間複雜度 (worst case time complexity) 以order 的方式表示。(8分) (注意：不可將此陣列數值進行排序，請加註解說明程式碼作法。)

### 試題評析

本題屬於二分搜尋法的變化題，測驗考生的程式能力。三元化搜尋使用兩個基準值mid1和mid2(其中，mid1<mid2)，對未排序資料進行partition，分成第1群：小於或等於mid1；第2群：大於mid1且小於mid2；第3群：大於或等於mid2。然後，遞迴進行搜尋。本題可以合併快速排序法中的partition處理與改寫binary search成為三元化搜尋，然後撰寫程式。

	本題可說是104年地特考古類似題。
<b>考點命中</b>	1. 《資料結構》，高點文化出版，王致強編著，頁9-23~9-25。 2. 《資料結構》，高點文化出版，王致強編著，頁10-7~10-8，精選例題3。

**答：**

(一)

```

class searching {
    private int A[100];
    int search(int A[], int x, int low, int high) {
        int pivot;
        // 如果搜尋範圍是空的傳回 0
        if (low>high) return 0;
        // 如果搜尋範圍只剩 1 項，檢查該項是否為 x
        if (low==high) {
            if (x==A[low])return 1;
            else return 0;
        }
        // 如果搜尋範圍不只 1 項，先整理好使得 A[low]<A[high]
        if (A[low]>A[high]) {
            swap(A[low],A[high]);
        }
        // 取A[low]為mid1，由左而右分群，第1群成為≤mid1；之後資料皆>mid1
        pivot=A[low];
        j=low;
        for (i=low+1; i<=high; i++) {
            if (A[i]<pivot) {
                j++;
                swap(A[i],A[j]);
            }
        }
        mid1=j;
        swap(A[low],A[j]);
        // 取A[high]做為mid2，將>mid1的資料，由右而左分群，第3群皆為≥mid2
        pivot=A[high];
        j=high;
        for (i=high-1; i>=mid1+1; i--) {
            if (A[i]>pivot) {
                j--;
                swap(A[i],A[j]);
            }
        }
    }
}

```

```

    }
    mid2=j;
    swap(A[high],A[j]);
    // recursive processing
    if (x<A[mid1]) search(A,x,low,mid1-1); // x 落在第 1 群, 遞迴處理
    else if (x<A[mid2]) search(A,x,mid1+1,mid2-1); // x 落在第 2 群, 遞迴處理
    else if (x>A[mid2]) search(A,x,mid2+1,high); // x 落在第 3 群, 遞迴處理
    else return 1; // 搜尋成功 x==A[mid1] 或 x==A[mid2]
}
}

```

(二) Worst Case Time :

$$T(n)=T(n-2)+O(n)$$

$$T(n)=O(n^2)$$

Best Case Time :

$$T(n)=O(n)$$

四、(一)請使用C語言寫一副程式void FindMeanAverage(int A [], int n, int \*mean, int \*average), 對一個未排序的 (unsorted) 且長度為n的陣列A[0:n-1], 尋找陣列中的中位數與平均數, 並分別存入mean及average運算複雜度。(17分) (二)請舉例說明此副程式最差情況 (worst case) 所花費的運算複雜度。(8分) (注意: 請加註解說明程式碼作法。)

<b>試題評析</b>	本題包含2個計算, (1)計算average平均值比較簡單; (2)計算中位數(median)須使用Quick Sort的變化應用: 找尋第k小元素(select kth small element)的方法來實作, 計算時, 取k=n/2即可。
<b>考點命中</b>	《資料結構》, 高點文化出版, 王致強編著, 頁9-33。

**答:**

(一)

```

void FindMeanAverage(int A[], int n, int *mean, int *average) {
    int i, sum, mid;
    for (i=0; i<n; i++) sum+=A[i];
    *average=sum/n;
    while (low<=high) {
        mid=partition(A, low, high);
        if (mid==n/2) {
            *mean=A[mid];
            break;
        }
        else if (mid>n/2) high=mid-1;
        else low=mid+1;
    }
}

```

```
    return;
}
int partition(int A[], int low, int high) {
    int pivot=A[low];
    int i, j=low;
    for (i=low+1; i<=high; i++) {
        if (A[i]<pivot) {
            j++;
            swap(A[i],A[j]);
        }
    }
    swap(A[low],A[j]);
    return j;
}
```

(二) Worst case time :

$$T(n)=T(n-1)+O(n)$$

$$T(n)=O(n^2)$$

高  
點  
·  
高  
上

【版權所有，重製必究！】

大吉

## 總複習班 → 提升統整力

- 求勝科目** 共同科目+專業科目
- 好試解籤** 重點歸納、時事修法以及命題趨勢提醒。
- 達人推薦** 張逸仙 普考地政

高點總複習課程不僅可以快速複習重點，命中率也很高！我特別推薦許文昌跟于俊明老師，教學認真、教材豐富，非本科系的考生也能快速上手，讀書更有效率！



三等 **5,000** 元 定價 8,000元起

四等 **4,000** 元起

大吉

## 題庫班 → 打造高分力

- 求勝科目** 經濟學/財政學/稅法/會計/審計/政會
- 好試解籤** 名師嚴選經典考題，傳授看題能力以及教導高分答題技巧！
- 達人推薦** 柯辰穎

高普考財稅行政雙榜  
隨著考期越來越近，我開始感到心慌，所以跑去報名會計&經濟&財政的題庫班，老師解題讓我釐清觀念，增加解題能力。



**1,800** 元起/科

4堂/科 定價 5,000元

# 高點 · 高上

## 高普考 衝刺

商資 · 地政 / 必勝錦囊

考運亨通

大吉

## 申論寫作班 → 論正寫題力

- 求勝科目** 審計/民法
- 好試解籤** 課前練題，高質量批改服務，建立答題架構，提高寫作高分力！
- 達人推薦** 李濤亦 高普考會計雙榜

高點老師請申論題命中率非常高！審計公報後期時間不太凶，只抓老師重點來背，申論竟拿到**32分**！



**2,500** 元/科

6堂起/科 定價 5,000元



大吉

## 公經進階班 → 鞏固強試力

- 好試解籤** 透析考題趨勢，加強進階內容，使考生能進一步掌握艱深考題。
- 達人推薦** 陳樂庭 高普考經建行政【狀元】

推薦張政(張家璋)老師的公經進階課程，他用數理詳細說明觀念，讓我實力大增！



**2,500** 元

以上考場優惠 110/10/20 前有效，限面授/VOD，當期最新優惠洽各分班櫃檯或高上生活圈！



另有**行動版課程**隨時可上  
試聽&購課，請至

1

知識達購課館  
ec.ibrain.com.tw



2

高點網路書店  
publish.get.com.tw

