GA Programming Assignment #5

1. Write a GA algorithm based on our slides for the 0-1 knapsack problem discussed in the class.
2. Develop a Java Program to implement your GA algorithm.
3. Use the following example to show your program working properly.

, this is the maximum weight of the objects that the knapsack can carry.

|  |  |  |
| --- | --- | --- |
| item | weight | profit |
| 1 | 3 | 9 |
| 2 | 2 | 5 |
| 3 | 5 | 14 |
| 4 | 4 | 11 |
| 5 | 6 | 16 |
| 6 | 3 | 8 |
| 7 | 4 | 13 |
| 8 | 1 | 3 |
| 9 | 7 | 15 |
| 10 | 8 | 23 |
| 11 | 9 | 29 |

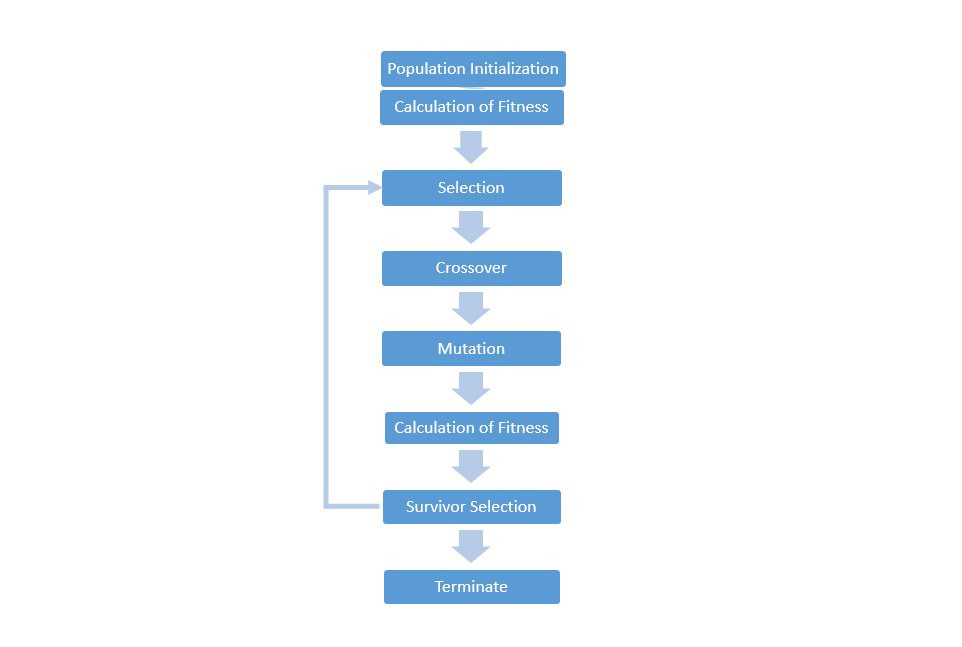
請繳交Java source code 和一份說明文件。The document must contain your algorithm and screenshots for running the example described in step 3。

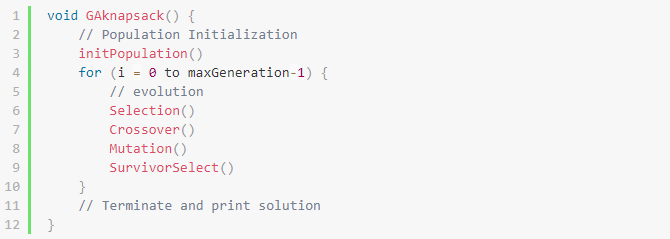
無說明文件，以零分計算此作業。

**基因演算法各步驟**

**架構**

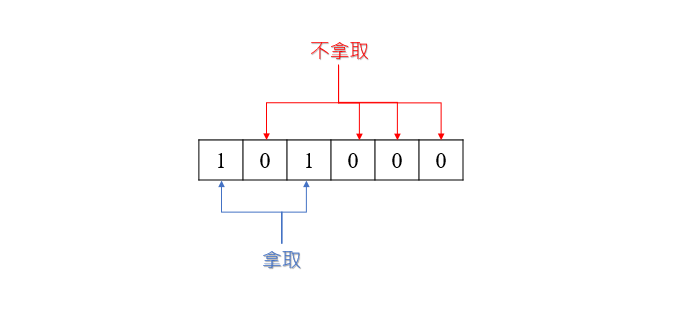
首先初始化 Population 的所有的個體，依據使用者輸入可以自由定義 Population 的大小。接著第二步驟計算每一個個體的 Fitness ，這邊的 Fitness 即為在背包限重內所獲得的總利益值。第三步驟開始進行演化首先使用輪盤法則挑選父代與母代，個體被選中的概率與其 Fitness 成正比，所以 Fitness 越大被選重的機率就越高。父代與母代經由輪盤法則挑選完畢後接著進行 Crossover，這邊是以單點交叉來實作。交叉完畢後最後會經由使用者所設定的突變率來判斷是否要對個體產生突變，突變的用意是為了避免最佳值落在區域最佳當中。演化完成後會把 Population 中比較優良的個體保留起來接著回到第二步驟計算 Fitness 並再一次新的演化直到使用者所設定的最大演化代數，或是在連續N代中最佳值都一樣，才停止迴圈並列印出結果。





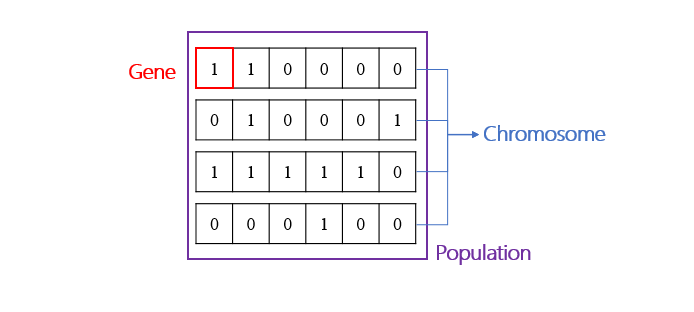
**Chromosome (染色體)**

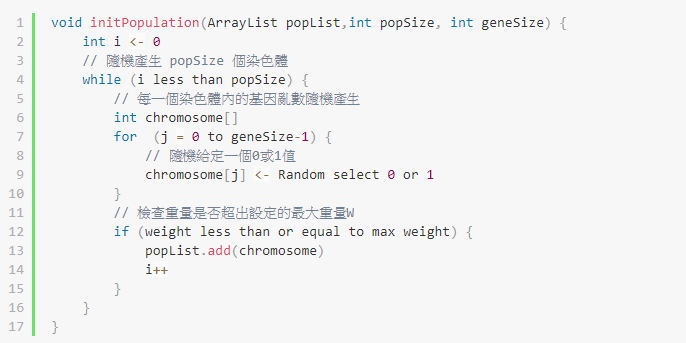
在01背包中物品的數量代表每個染色體(Chromosome)的長度，每一個基因以0或1表示1代表該物品要拿;反之0代表該物品不拿。下圖範例為一個 Chromosome 內部含有六個基因，每一個基因代表每一個物品要不要拿。



**Population Initialization**

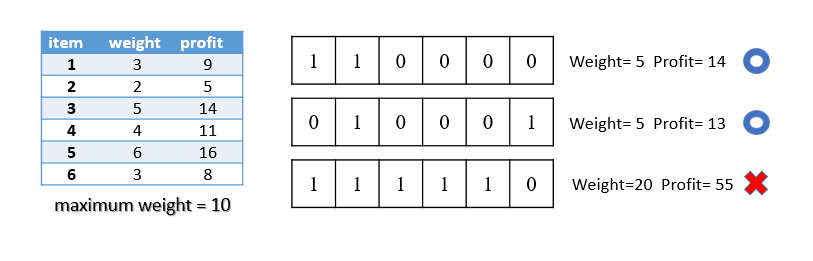
初始化的方式為建立N個染色體個體並隨機用亂數將每個基因賦予初始值，建立好之後會檢查該筆染色體的組合是否超出設定的最大重量(W)，若超重則拋棄並重新亂數產生出新的一條染色體;直到數量到達使用者所設定的 Population 的大小即結束初始化。

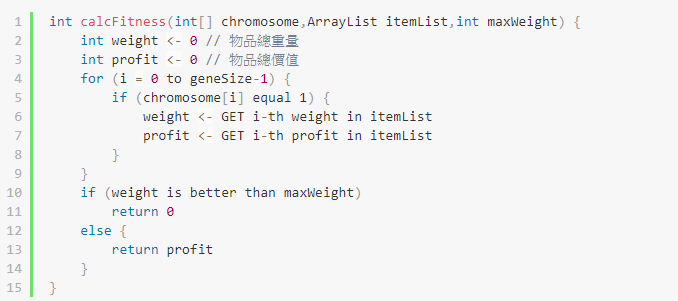




**Calculation of Fitness**

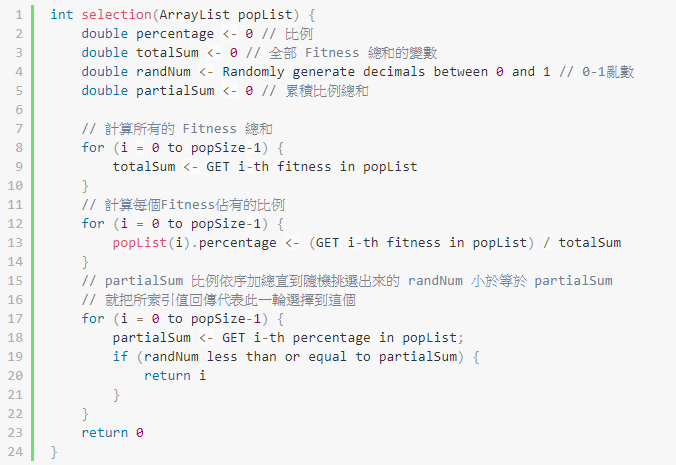
在每次繁殖過程中需要計算每個染色體的Fitness(適應值)，在每個染色體的基因串列中1代表該物品要拿，所以就將該物品的重量(Weight)和利益值(Profit)分別記錄並累加起來。加總完成後檢查該染色體基因組合的背包重量是否超出設定的最大重量 maxWeight，若超重則回傳0;反之回傳總利益值做為該染色體基因組合的Fitness。

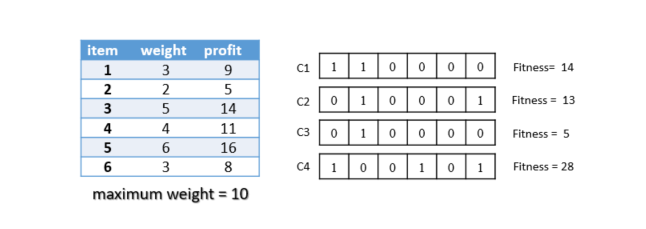
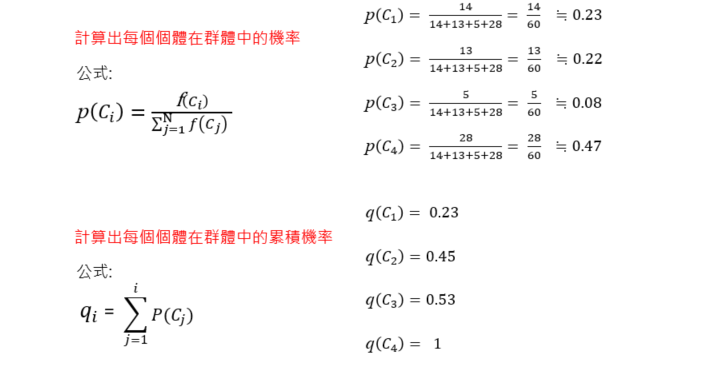
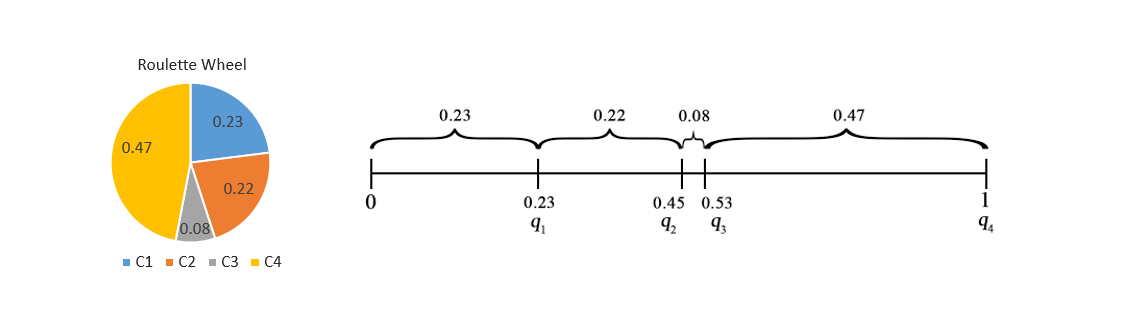




**Selection**

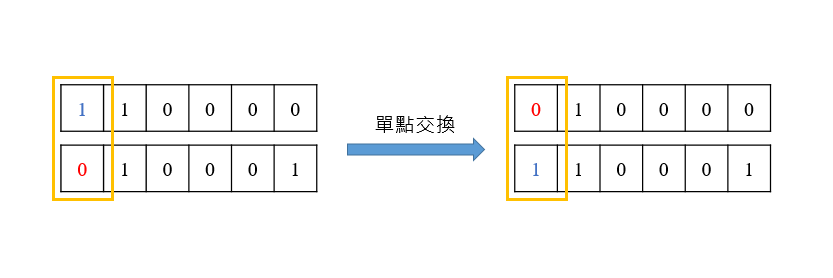
選擇父代與母代染色體來產生下一代，這邊選擇的方式使用 輪盤選擇法 (Roulette Wheel Selection) 來實作。所謂的輪盤選擇法就是在整個族群中，每個個體存活下來或是可以產生後代的機率和個體分數成正比。也就是說 Fintness 越大的個體存活下來被選擇到的機率就越大，但弱勢個體也有存活的可能，不是絕對的淘汰。就想像個體放在飛標靶上，而個體分數就對應到個體擁有的標靶面積。

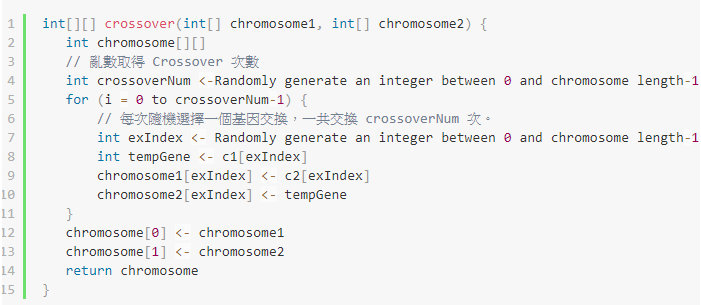


**Crossover**

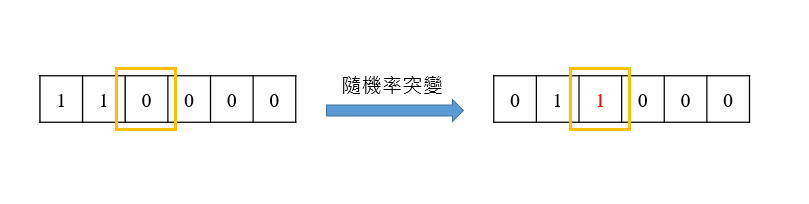
所謂 Crossover 是指對兩個(父、母代)配對的染色體相互交換其部分基因，從而形成新的個體。在這邊是使用隨機單點基因交換，並且隨機交換 N 次。

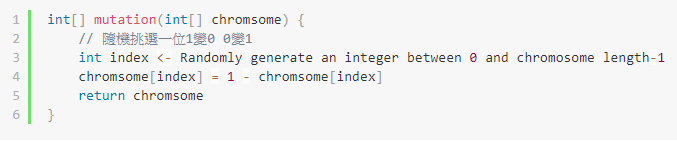




**Mutation**

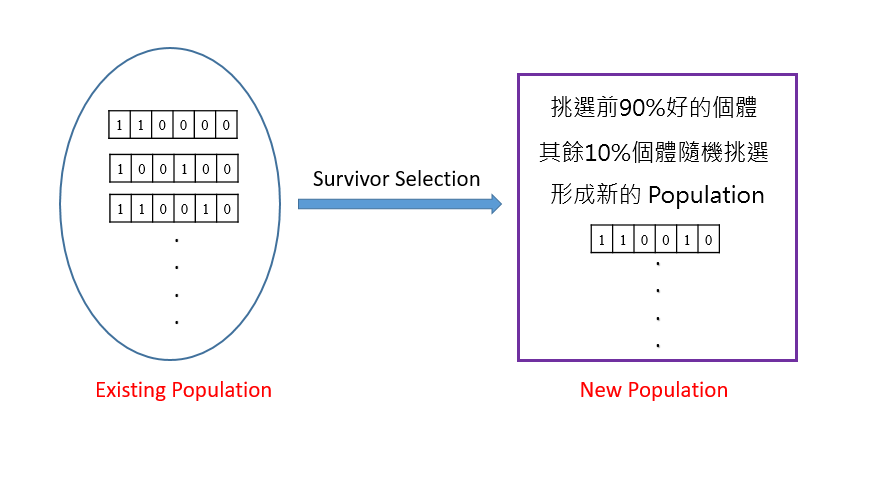
在演化的過程中有一定的機率突變，突變的時機在當父母繁殖 Crossover 後產生出來的新子代，此時新的子代經由突變率會有一定的機率突變。這裡突變的方式是隨機將某一個基因做交換(0變1;1變0)。

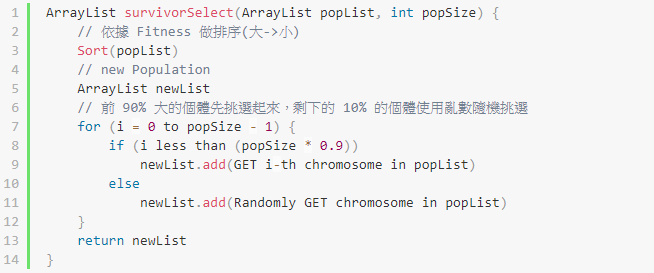




**Survivor Selection**

每一代演化後要保留比較好的個體作為下一次演化的 Population ，這裏挑選的方式是。首先排序按照 Fitness 數值由小到大排列，接著從裡面挑選前 90% 大的個體先挑選起來，剩下的 10% 的個體使用亂數隨機挑選。這麼做的原因是要確保我們的最佳解不會落在區域最佳。

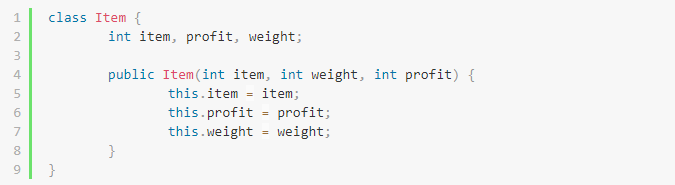




**程式碼解說**

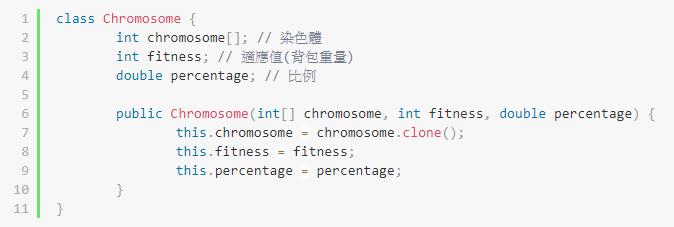
**Item 類別**

此類別建立是存放的所有的物品，分別記錄著 item 物品編號、profit 物品的利益值、weight物品的重量。以及建立一個建構子每一次新增一個物品時即初始化並放入 ArrayList 串列中儲存。



**Chromosome 類別**

此類別是紀錄染色體的每個基因，總共有三個變數分別為 chromosome[] 存放染色體基因的陣列，這裡是以整數型態的陣列從放基因;接著是 fitness 適應值變數，此變數是存放每一組染色體的 profit 值;最後一個變數是 percentage 比例，此變數記錄著該染色體在目前群體中所佔的比例是多少。最後一樣建立建構子當產生新的染色體時並初始化上述所有的變數。

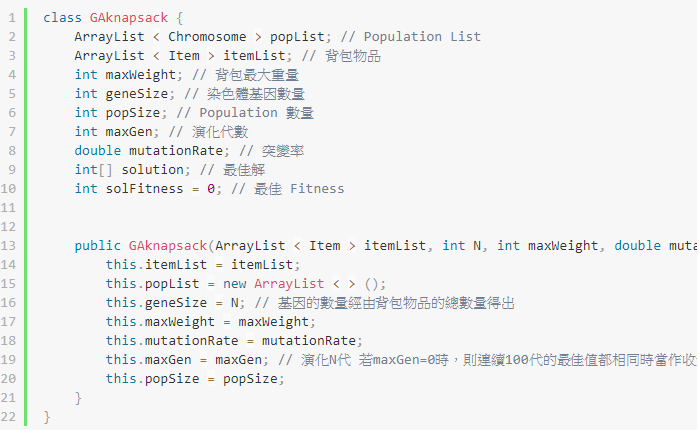


**GAknapsack 類別**

此類別是 Genetic Algorithm 的實例，分別進行Population的初始化以及選擇->交配->突變。由於此類別比較複雜故將每一個 Function 拆開逐一來解釋。

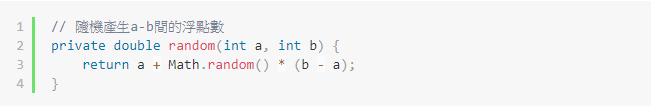
**全域變數**

在此類別中分別建立以下變數。第一個為 Population List 使用 ArrayList 儲存每一個 Chromosome(染色體)。第二個為背包的物品依樣是使用 ArrayList 來儲存每一個 Item。第三個為背包的最大重量 maxWeight 。第四個是染色體基因數量 geneSize。第五個是Population 數量 popSize。第六個是突變率 mutationRate。第七個是最佳解 solution[] 此變數是一個整數的陣列型態記錄著每次演化後的最佳的一組解(染色體)。最後一個變數為 solFitness 此變數記錄著最佳解的 Profit 背包物品的利益值。



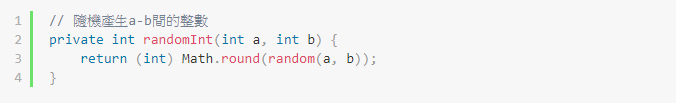
**random() 函式**

此函式是利用 Math 函式庫中的 random() 函式來隨機產生a-b間的浮點數，此函式擁有兩個傳入值分別為a(起始值)與b(最終值)，最後將隨機產生出來的浮點數回傳。



**randomInt() 函式**

此函式跟上述的 random() 函式類似只不過是將回傳值強制轉型成整數(int)型態。此函式擁有兩個傳入值分別為a(起始值)與b(最終值)，最後將隨機產生出來的整數回傳。



**initPopulation() 函式**

基因演算法第一步驟是初始化 Population (隨機產生子代)，傳入值分別有 Population 大小popSize，以及染色體基因的長度 geneSize。

初始化的方式為建立 popSize 個染色體個體並隨機用亂數將每個基因賦予初始值(程式6-9行)，建立好之後會呼叫 calcFitness() 計算 Fitness 並檢查該筆染色體的組合是否超出設定的最大重量(maxWeight)，若超重則 calcFitness() 函式會回傳 0 因此不將此組染色體放入 Population List 當中並重新亂數產生出新的一條染色體;直到數量到達使用者所設定的 Population 的大小即結束初始化。



**calcFitness() 函式**

在每次繁殖過程中需要計算每個染色體的Fitness(適應值)，在每個染色體的基因串列中1代表該物品要拿，所以就將該物品的重量(Weight)和利益值(Profit)分別記錄並累加起來(程式5-10行)。加總完成後檢查該染色體基因組合的背包重量是否超出設定的最大重量 maxWeight(程式12-16行)，若超重則回傳0;反之回傳總利益值做為該染色體基因組合的 Fitness。



**selection() 函式**

選擇父代與母代染色體來產生下一代，這邊選擇的方式使用 輪盤選擇法 (Roulette Wheel Selection) 來實作。所謂的輪盤選擇法就是在整個族群中，每個個體存活下來或是可以產生後代的機率和個體分數成正比。也就是說 Fintness 越大的個體存活下來被選擇到的機率就越大。

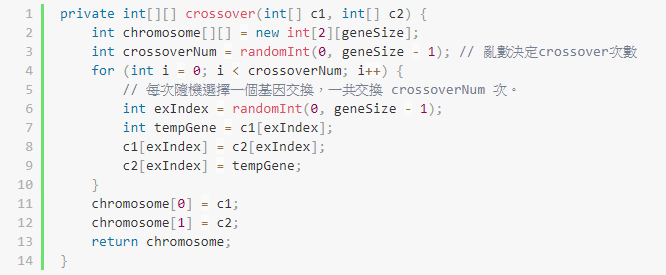
首先計算所有的染色體 Fitness 總和並且儲存在 totalSum 變數當中(程式9-11行)，接著在分別計算每個染色體在整個群體中所佔有的比例，計算的方式為(目前某一個染色體的Fitness)/totalSum (程式13-16行)。全部的染色體都計算好後接著計算每個Fitness佔有的比例(程式13-16行)。最後依據隨機產撐出來的變數 randNum 產生 0-1 之間的小數，進行每一個 Fitness 比例的累加直到隨機挑選出來的 randNum 小於等於 partialSum 就把目前的所索引值回傳代表此一輪選擇到這個。



**crossover() 函式**

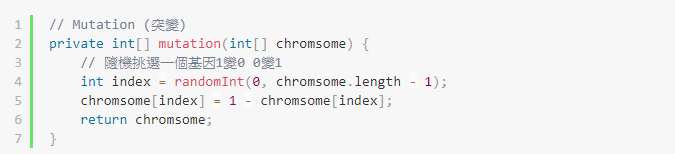
所謂 Crossover 是指對兩個(父、母代)配對的染色體相互交換其部分基因，從而形成新的個體。在這邊是使用隨機單點基因交換，並且隨機交換 N 次。

此函式會有兩個參數 c1、c2 分別代表父代與母代，首先會有一個 crossoverNum 變數進行亂數初始化，接著進入迴圈每次隨機選擇一個基因交換，一共交換 crossoverNum 次，並且每一次做單點的基因交換。



**mutation() 函式**

在演化的過程中有一定的機率突變，突變的時機在當父母繁殖 Crossover 後產生出來的新子代，此時新的子代經由突變率會有一定的機率突變。這裡突變的方式是隨機將某一個基因做交換(0變1;1變0)。



**evolution() 函式**

此函式為 repopulation 實例進行演化，分別處理 Select (程式11-18行) 進行輪盤選擇法挑選父代與母代，接著進行 Crossover 隨機做單點交換(程式10-13行)，交叉完成後隨著突變率有一定的機率突變(程式15-18行)。上述三個動作都完成後進行 Fitness 計算並檢查此染色體是否超重，若沒超重則將染色體放回去 Population List 中(程式碼23-26)。

最後每一代演化後要保留比較好的個體作為下一次演化的 Population ，這裏挑選的方式是。首先排序按照 Fitness 數值由小到大排列，接著從裡面挑選前 90% 大的個體先挑選起來，剩下的 10% 的個體使用亂數隨機挑選。這麼做的原因是要確保我們的最佳解不會落在區域最佳(程式碼37-45)。



**mergeSort() 排序函式**

由於每次演化後要挑選比較好的子代做為下一次演化的依據，所以我們要先將每個染色體內的 Fitness 由大到小做排序，這邊排序是使用 Merge Sort 合併排序實作。



**GArun() 函式**

此函式是主程式中被呼叫基因演算法執行的函式，首先會進行 Population 初始化動作(程式第4行)。接著程式6-13行進行演化選擇->交配->突變 (循環重複到 maxGen 代)，此外若 maxGen=0 時，則會檢查連續100代的最佳值都相同時當作收斂故跳出迴圈。

最後是輸出結果，會顯示此次GA執行的參數設定，這裡會依序列出群體大小、突變率、以及終止條件(演化最大代數)。接著印出GA後的結果第一行為此次執行最佳的一組染色體(Chromosome);接著輸出最佳解的背包物品組合;最後則是背包內商品的最大利益值。



**main() 主函式**

在主函式中首先要求使用者輸入背包內容，輸入的第一行程式會要求使用者輸入物品數量(N)。第二行輸入最大的背包重量上限(maxWeight)。接著第三行開始會有連續N筆資料輸入，每一行請輸入兩個數值;第一個為物品的重量(weight)、第二個數為此物品的利益值(profit)。

接著程式第23行呼叫 GAknapsack 類別變數並初始化執行基因演算法，參數分別放入物品、物品數量、背包最大承重、突變率、演化代數、Population 數量。其中突變率、演化代數、Population 數量可以依照使用者喜好手動調整參數，若演化代數輸入0則終止條件會檢查連續100代的最佳值都相同時當作收斂故跳出迴圈並列印出結果。



**測試**

Input:  
輸入的第一行程式會要求使用者輸入物品數量(N)。第二行輸入最大的背包重量上限(maxWeight)。接著第三行開始會有連續N筆資料輸入，每一行請輸入兩個數值;第一個為物品的重量(weight)、第二個數為此物品的利益值(profit)。

Output:  
首先會依序印出所有物品的重量(weight)與利益值(profit);接著會顯示此次GA執行的參數設定(若要更改參數要在程式裡做修改)，這裡會依序列出群體大小、突變率、以及終止條件(演化最大代數)。演化結束後會印出GA後的結果第一行為此次執行最佳的一組染色體(Chromosome);接著輸出最佳解的背包物品組合;最後則是背包內商品的最大利益值。

* 測試1

測資:

11

30

3 9

2 5

5 14

4 11

6 16

3 8

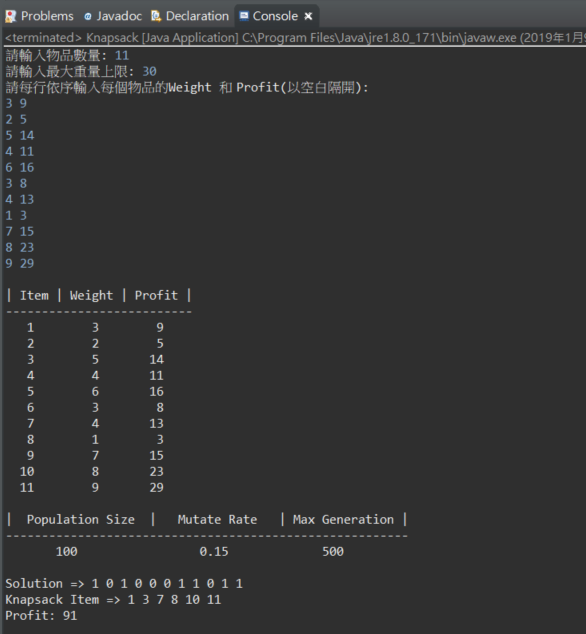
4 13

1 3

7 15

8 23

9 29



Profit: 9+14+13+3+23+29 = 91

* 測試2

測資:

20

6844

1999 1

901 4

2227 5

532 4

3554 4

1815 5

3292 2

3093 4

2442 3

393 2

3657 5

3694 4

2283 5

366 5

3626 2

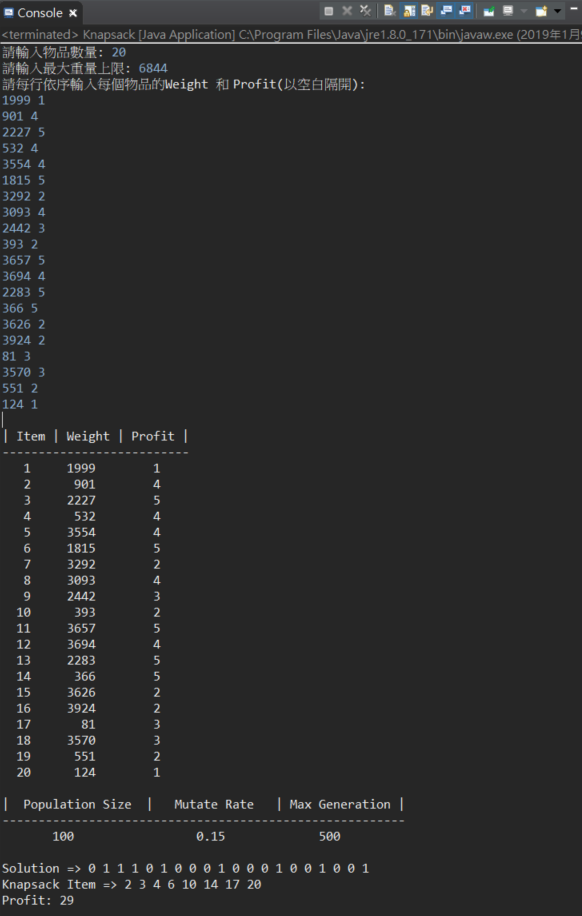
3924 2

81 3

3570 3

551 2

124 1



Profit: 4+5+4+5+2+5+3+1 = 29

* 測試三

測資:

6

30

15 15

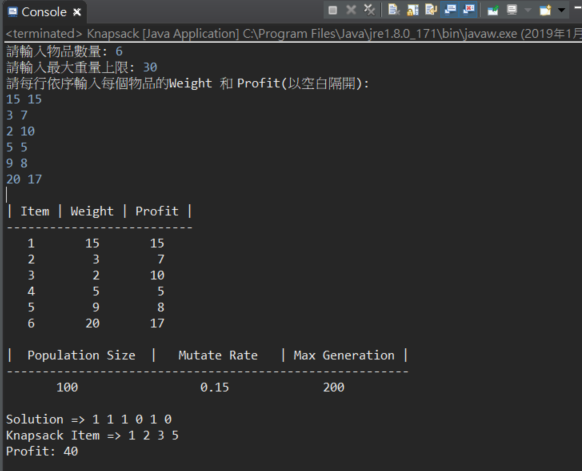
3 7

2 10

5 5

9 8

20 17



Profit: 15+7+10+8 = 40