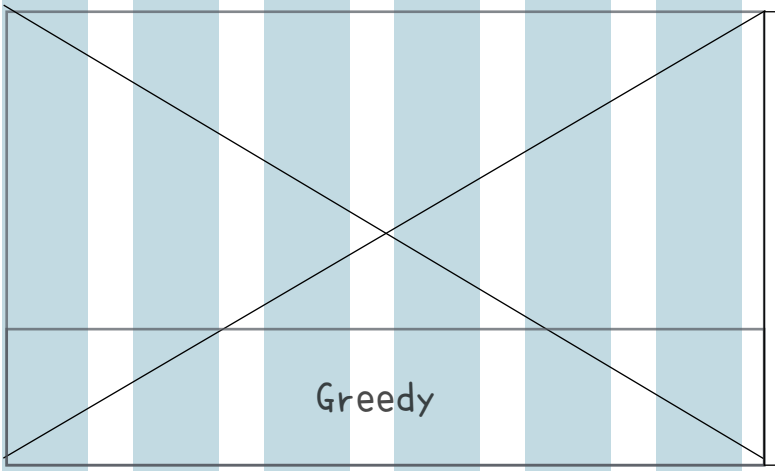
分類題庫



Greedy(貪心法;貪婪算法)

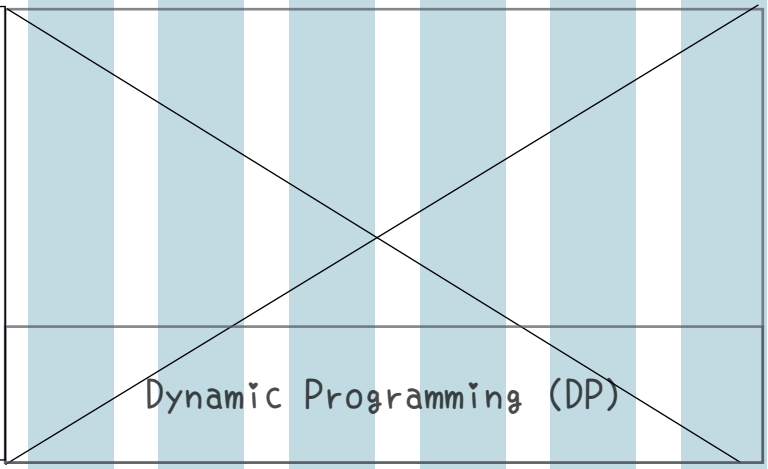
在每一步選擇中都採取在當前狀態下最好或最優（即最有利）的選擇，從而希望導致結果是最好或最優的算法。比如在旅行推銷員問題中，如果旅行員每次都選擇最近的城市，那這就是一種貪心算法。

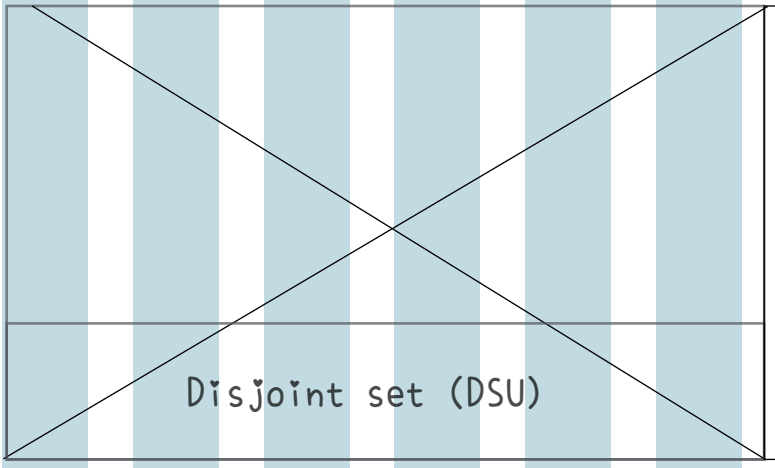
在有最優子結構的問題中尤為有效。最優子結構的意思是局部最優解能決定全局最優解。子問題的最優解能遞推到最終問題的最優解。

Dynamic Programming（動態規劃）

通過把原問題分解為相對簡單的子問題的方式求解複雜問題的方法，常適用於有重疊子問題和最優子結構性質的問題。

為減少計算量，一旦某個給定子問題的解已經算出，則將其記憶化存儲，以便下次需要同一個子問題解之時直接查表。這種做法在重複子問題的數目關於輸入的規模呈指數增長時特別有用。





Disjoint set（互斥集;並查集）

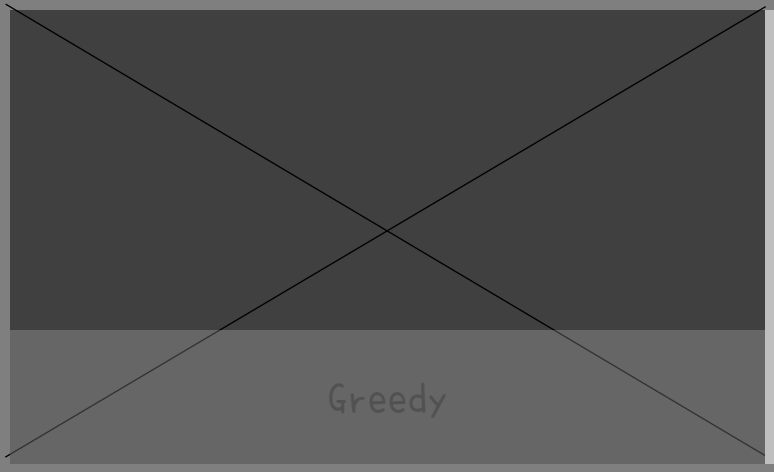
一種樹型的資料結構，其保持著用於處理一些互斥集合（Disjoint Sets）的合併及查詢問題。聯合-查找算法（union-find algorithm）定義了兩個操作用於此資料結構：

- Find：確定元素屬於哪一個子集。它可以被用來確定兩個元素是否屬於同一子集。
- Union：將兩個子集合併成同一個集合。

copyright©NTUT-ACM

Tel:09xx-xxx-xxx(XXX)

分類題庫

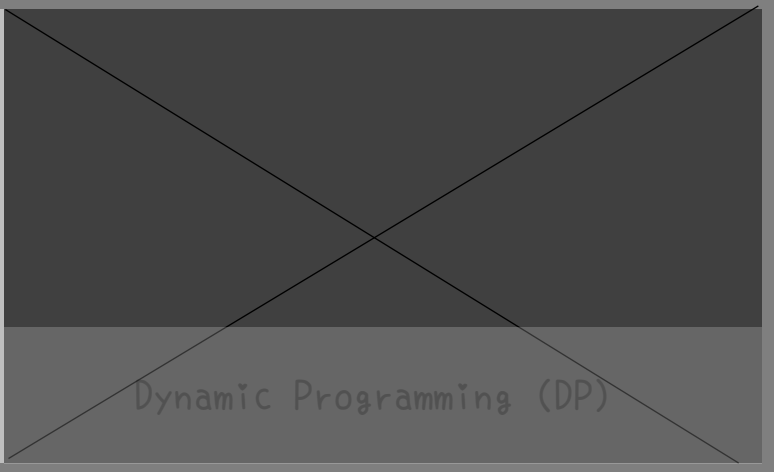


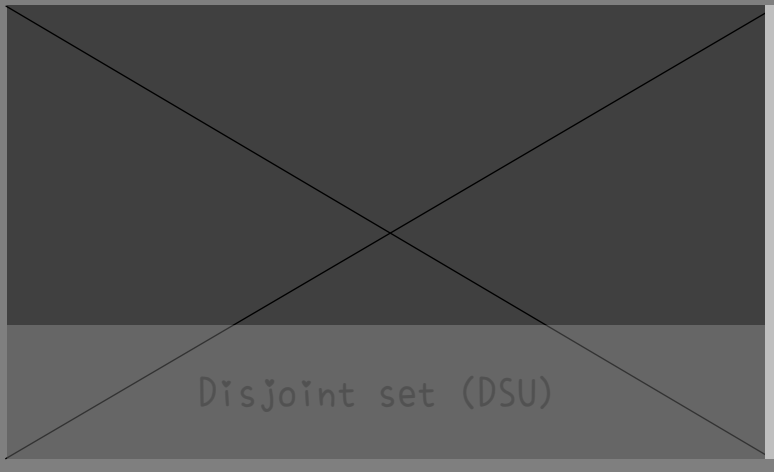
Greedy(貪心法;貪婪算法)

在每一步選擇中都採取在當前狀態下最好或最優（即最有利）的選擇，從而希望導致結果是最好或最優的算法。
在有最優子結構的問題中尤為有效。最優子結構的意思是局部最優解能決定全局最優解。子問題的最優解能遞推到最終問題的最優解。

Dynamic Programming（動態規劃）

通過把原問題分解為相對簡單的子問題的方式求解複雜問題的方法，常適用於有重疊子問題和最優子結構性質的問題。
為減少計算量，一旦某個給定子問題的解已經算出，則將其記憶化存儲，以便下次需要同一個子問題解之時直接查表。這種做法在重複子問題的數目關於輸入的規模呈指數增長時特別有用。





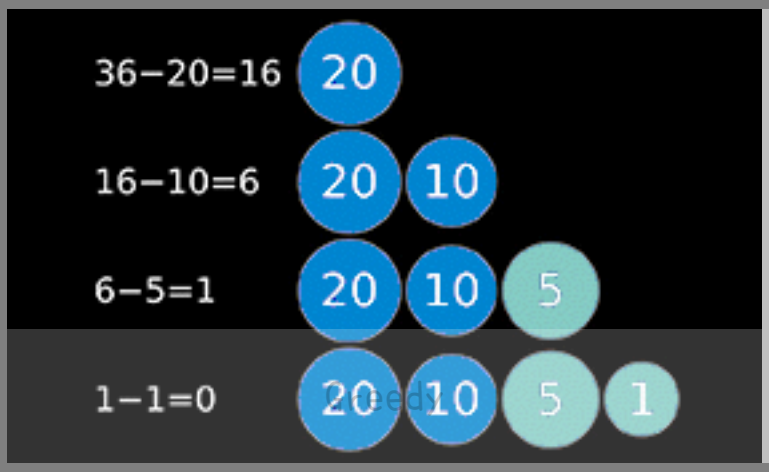
Disjoint set（互斥集;並查集）

一種樹型的資料結構，其保持著用於處理一些互斥集合（Disjoint Sets）的合併及查詢問題。聯合-查找算法（union-find algorithm）定義了兩個操作用於此資料結構：

- Find：確定元素屬於哪一個子集。它可以被用來確定兩個元素是否屬於同一子集。
- Union：將兩個子集合併成同一個集合。

copyright©NTUT-ACM
Tel:09xx-xxx-xxx(XXX)

分類題庫

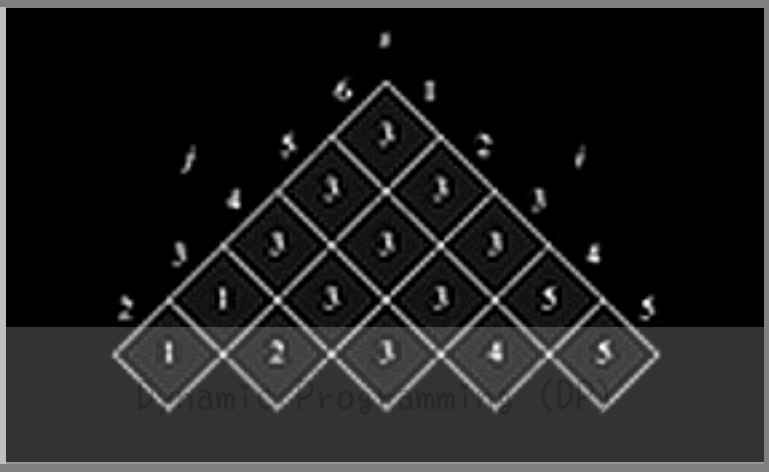


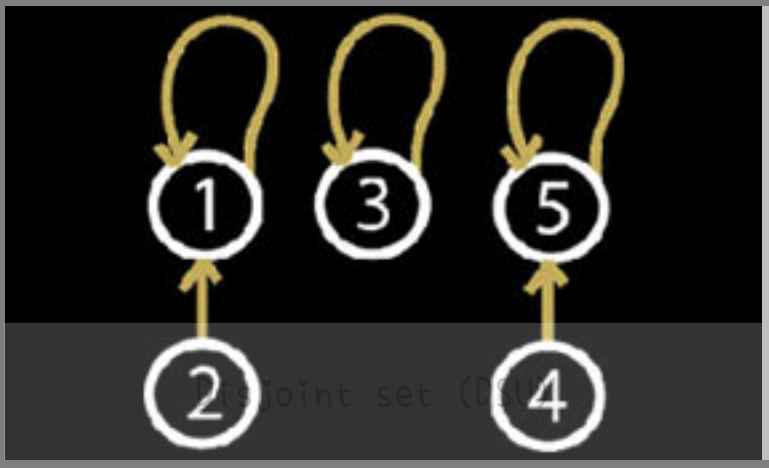
Greedy(貪心法;貪婪算法)

在每一步選擇中都採取在當前狀態下最好或最優（即最有利）的選擇，從而希望導致結果是最好或最優的算法。
在有最優子結構的問題中尤為有效。最優子結構的意思是局部最優解能決定全局最優解。子問題的最優解能遞推到最終問題的最優解。

Dynamic Programming（動態規劃）

通過把原問題分解為相對簡單的子問題的方式求解複雜問題的方法，常適用於有重疊子問題和最優子結構性質的問題。
為減少計算量，一旦某個給定子問題的解已經算出，則將其記憶化存儲，以便下次需要同一個子問題解之時直接查表。這種做法在重複子問題的數目關於輸入的規模呈指數增長時特別有用。





Disjoint set（互斥集;並查集）

一種樹型的資料結構，其保持著用於處理一些互斥集合（Disjoint Sets）的合併及查詢問題。聯合-查找算法（union-find algorithm）定義了兩個操作用於此資料結構：

- Find：確定元素屬於哪一個子集。它可以被用來確定兩個元素是否屬於同一子集。
- Union：將兩個子集合併成同一個集合。