**A. 證明二元搜尋演算法對於只考慮比較的搜尋演算法是最佳的。**輸入:n的排序好的數字  
每次比較完刪掉一半的數字,直到只剩一個數字=>n/2k = 1, k = logn = Ω(logn)  
二元搜尋演算法的time complexity = O(logn), 因此為最佳

**B. 證明歐基里德最小含括樹( Euclidean minimal spanning tree problem)問題的問題下界為Ω(n log n)。**排序問題reduce to Euclidean minimal spanning tree problem  
輸入:x1 ~ xn -> (x1,0 ~ xn,0)  
經由Euclidean minimal spanning tree解完後，因為等同一維座標(全部點共線)，可知樹會從x小的連到x大的  
因此輸出轉回時,x1 ~ xn會是排序好的  
已知排序下界為Ω(nlogn)  
變轉時間為O(n)  
因此可知Euclidean minimal spanning tree problem下界為Ω(nlogn)  
http://www-ma2.upc.es/vera/cotasnuevas.pdf(參考:第五題)

**C. 如何證明一個問題是NP-hard?**每個NP問題都可以多項式時間變轉為此問題  
已知∀y, y∈NP , y∝SAT , reduce具有遞移性  
因此 若此問題x, SAT∝x , 則∀y, y∈NP , y∝x , x就為NP-hard

**D. 設計NP演算法解決集合覆蓋(set covering)決策問題。**輸入:一個宇集U和其子集F所形成之集合  
輸出:具有一個集合S由F組成, 其聯集為U,且F的數量小於等於常數C(最小) success 反之failure  
S←choice({F1, F2, F3, F4, F5…}//從集合中任意猜測c個成員  
if S的成員聯集為U , then success  
 else failure

**E. 以下為分割問題(partition problem)的描述：給定一個正整數集合S，是否存在一種分割將S分割為S1與S2集合，且S1集合的元素加總值與S2相同？證明此分割問題為一NP 問題。**  
若此問題可以非確定性演算法解出即為NP問題  
輸入:一個正整數集合S, set P[]儲存分割的元素(元素量從1~n-1)  
輸出:存在分割的集合S1、S2元素加總值相同success 反之failure  
for i=1 to n-1  
 P[i] = choice({S})//猜測1~n-1個元素

End for  
if 任意P[i] 其元素總和 = S元素總合的一半 success  
 若全部P[i]皆不符合 failure

**F. 已知漢米爾頓迴路(Hamiltonian circuit)決策問題為NPC，證明旅行推銷員(Traveling salesperson)決策問題也是NPC。**證明其為NP:  
輸入:一個集合S包含平面上n個點  
輸出:有一個封閉旅途,每個點只經過一次,其長度小於等於L  
 存在success, 反之failure  
 p = choice(S)//選擇起點  
for i=0 to n-1  
 next = choice(S)//選擇還未選過的點當下一點,最後選的一點為一開始起點  
 D += d(p,next)  
 p = next  
end for  
  
if(D<=L) success  
else failure

證明其為NP-hard:漢米爾頓迴路∝TSP(給定一起始點的TSP問題)

**G. 如何可以證明P<>NP(P不等於NP)?如何可以證明P=NP?(需要解釋你的答案)**1. P<>NP: if SAT∉P(SAT 無法用確定性演算法在多項式時間內解出)  
 則NP≠P

因為∀y, y∈NP , y∝SAT, 若SAT∉P, 則所有NP問題都可以變轉為SAT無法用確定性演算法在多項式時間內解出  
2.P=NP: if SAT∈P(SAT 可用確定性演算法在多項式時間內解出)

則P=NP

因為∀y, y∈NP , y∝SAT, 若SAT∈P, 則所有NP問題都可以變轉為SAT用確定性 演算法在多項式時間內解出

**H. 說明在n個數值中找出第二大的數至少需要(n-2+**└logn┘**)的比較次數。**step1:找最大:n-1  
step2:找第二大(跟之前比過的比): └logn┘(層數)-1  
總共:n-1+└logn┘-1 = n-2+logn