最短路演算法

(1) 最短路徑的最佳子結構

假設路徑 p = {v1, v2, v3, … vk}為 v1 到 vk 的最短路徑，對於任何在此路徑上的 i 和 j，路徑 p_{ij} = {vi, …vj}必為一條從 vi 到 vj 的最短路徑

證明：將路徑 p 分解成 v1 ~ vi ~ vj ~vk，假設路徑 p 的長度 w(p) = w(v_{1j}) + w(p_{ij}) + w(p_{jk})，若 vi 和 vj 存在一條不在 p 上的路徑 p_{ij}’ < w(p_{ij})，那代表 v1 和 vk 之間存在更短的路徑，跟一開始的假設矛盾。得證。

(2) 鬆馳(Relax)操作

假設 d[i] 為原點到 v_i 的最短矩離，則定義鬆馳操作為：

```
RELAX(u, v, w){
    if   d[v] > d[u] + w(u, v)
    then d[v] = d[u] + w(u, v)
}
```

Layout(POJ 3169)

農夫約養了 N 頭牛，編號分別是 1 到 N。現在他們要進食，按照編號順序排成了一排。在它們之間有一些牛關係比較好，所以希望彼此之間不超過一定距離，也有一些牛關係比較不好，所以希望彼此之間至少要相隔一定距離。此外，牛的性格比較機車，所以可能有多頭牛擠在同一個位上。給出了 ML 個關係好的牛的關係(AL, BL, DL)，代表牛 AL 之間與牛 BL 之間不能超過 DL；同時給出 MD 個關係不好的牛的關係(AD, BD, DD)，表示牛 AD 和牛 BD 之間距離不能小於 DD。在滿足上述條件的情況，求出 1 號牛和 N 號牛之間的最大距離。如果不存在任何一種排列方法滿足條件則輸出-1。無限大的情況輸出-2