PA2 report

B04505001 工海三 周明德

**Prove the Optimal Substructure:** Write a recursive function for this problem. Prove that this problem has *optimal substructure* so that it can be solved by dynamic programming. Highly recommended to use the following hint while designing your optimization tool.

|  |
| --- |
| **Hints: (it is NOT required to use this model)**  Suppose *d[i]* means the shortest distance after picking up the *ith* food, and the maximum capacity is *C*. *dist2origin(i)* means thedistance between the *ith* food and the origin. *dist(a, b)* means the distance you need to travel between the *ath* food and the *bth* food. We can find the formula (please complete it):  Assume that *j* was the **second-last** food ID (the last food ID is *i*) after which you return to the formicary. So *j* is smaller than *i*. We also want to make sure that the sum of weight from *j+1* to *i* no larger than *C*. Please notice that if the minimum value of *d[i]* is found when *j = 0*, it means the ant should pick up the food from the *1st* to the *ith* and then go back to the formicary only one time. *d[0]* should be zero to make the formula reasonable under this condition. The following are the constraints corresponding to above formula: |

Suppose d[i] means the shortest distance after picking up *ith* food and carry it back to the formicary.

If d[k] is a sub problem in d[i], and d[k] is not the shortest distance after picking up the k*th* food.

Replace d[k] with the shortest distance after picking up the k*th* food and carry it back to the formicary, then d[i] will be less than it was. This violate assumption that d[i] is the shortest distance after picking up *ith* food and carry it back to the formicary.

Proof by contradiction. This problem has *optimal substructure.*