1、

可以从最初空手状态开始进行深度优先搜索。每次搜索都可以买一个大礼包或者按原价全部购买。 直到买到指定数目的商品为止就可以得到一个可行解。采用爬山策略，优先搜索省钱的购买方式。当得到可行解时，以该最小可行解对其他分支进行剪枝。

伪代码：

def shoppingOffers(self, price, special, needs):

for i in special:

compute the money that saved t1

sav.append([i,t1])

sav.sort()

parm ←[0] \* len(price)

stk ← [[parm, 0]]

min\_cost←0xffffffff

while True do

p1←stk[-1][0]

p2←stk[-1][1]

del(stk[-1])

if p1==needs then

min\_cost←p2

if stk is empty then

return min\_cost

copy p1 to np1

copy p2 to np2

comput the new cost np2 after buying the remain goods without gift

if cost<min\_cost then

stk.append([needs,np2])

for i in sav:

copy p1 to np1

np2=p2

comput the new number of goods np1 after buying the remain goods without gift

if the np1 is not beyond needs after buying i then

comput the new cost np2 after buying the remain goods without gift

if the cost after buying i<min\_cost then

stk.append([np1,np2])

if stk is empty then

return min\_cost

2、

（1）

g(n)为沿着当前路径从起点到n点的路径长度。

h\*(n)为n点到可拓展的点的最短边长。（所谓可拓展点是不会导致部分成环的点）

（2）

