* 1. E
  2. A
  3. C
  4. C
  5. D
  6. A
  7. D
  8. A
  9. B
  10. D

2.1

For the Cashier’s algorithm, since the target value is 104, so each time we pick the largest one from the set, which will give us:

100 + 1 + 1 + 1 + 1

5 coins in total.

2.2

For best solution:

70 + 34

2 coins in total.

3.1

For profit 50, we can use item 4+5 which will give us 50 profits and weight 13.

3.2

We can design the algorithm by, what is the minimum space needed to make X value with the first k items.

Text

Description automatically generated

3.3

If W=5, the solution with the largest quality would be:

Item 3, since the v to Alice and v to Bob are both 18, so the quality would be 18 in this case.

3.4

* Find a rate by calculate v’ = min(v-Alice, v-Bob) / w
* Try to put the most promising v’ into the bag

4.1

For (1,2), C1=1, C2=3, Value = 5C1+4C2 which is V=5X1 + 4X3, Finally, V = 17

For (2,1), C2=2, C1=3, Value = 5C1+4C2 which is V=5X3 + 4X2, Finally, V = 23

Thus, (1,2) has smaller value

4.2

Since w1/p1 > w2/p2, so (2,1) will have a smaller value.

4.3

For (2,3,4,1,5), V=5C1+4C2+3C3+4C4+5C5=5\*10+4\*2+3\*5+4\*9+5\*15=184

For (2,3,4,5,1), V=5C1+4C2+3C3+4C4+5C5=5\*15+4\*2+3\*5+4\*9+5\*14=204

Thus, (2,3,4,1,5) has smaller value

4.4

Based on previous calculation, will has a smaller value.

4.5

The solution is (n, n-1, … , 2, 1)

5.1

For the greedy algorithm:

* build the map
* find the depth for each node
* try to get the highest depth of the node
* set up the course with no prerequisite

5.2

Top-down dynamic programming design:

* setting prerequisites
* add the course into bucket if the course has not taken and its prerequisites are already finished.
* If bucket with available courses is greater than k then find all possible combination and get minimal number of semesters to complete rest of them.
* If can take all the courses in one semester then do it.