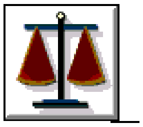
Design and Analysis of Algorithms



Problem #1. *A "true" medieval story…*

Sir Arthur de Templar was paid 27 gold coins for the information he provided on the last known location of the Holy Grail. Soon, from the highly trusted sources, he has learned that one of the 27 gold coins is not fully gold (it has some admixture of iron and copper and hence its weight differs (either heavier or lighter) from the weight of the other 26 true gold coins). Sir Arthur paid a visit to his closest friend drug-maker Mr. Drugless and in a few seconds they identified the fake coin. Legend says that they used just an old weighing device of Mr. Drugless (depicted on the Figure above) and could identify the fake coin in only 4 weighings.

1. Repeat the procedure used by Sir Arthur de Templar and Mr. Drugless for identifying the fake coin among 27 gold coins in just 4 weighings.
2. Extend it to an algorithm for identifying a single fake coin among ***N*** given gold coins using minimum number of weighings. How many weighings do you need?

(A weighing is an operation of putting *k* coins on one plate of the device and *k* other coins on the other plate (*k=1 or 2 or 3 …*) and checking if the device is in equilibrium, and if not, which plate is heavier, lighter.)

Problem #2.

Demonstrate the insertion of the keys 4, 27, 18, 14, 19, 32, 11, 16 into a hash table with collisions resolved by *linear probing*. Assume that the table has 9 slots and that the hash function is ***h(k) = k mod 9***. Draw the state of the hash table after all insertions.

What strategies do you know other than *linear probing* for handling *collisions* in a hash table?

Problem #3.

The following is a road distances chart that connects several important cities. All roads in a region are damaged due to a catastrophe. An emergency crew can repair only 10 miles of road in a day. We need to find the roads to be repaired urgently to restore the connectivity between the cities (i.e., at least one way exists to get from any city to any other city) within a minimum number of days. What algorithm will you be applying? Show step-by-step how your algorithm will find the solution for the listed cities. What is the run time complexity of your algorithm?

	Kenton	Akronia	Leveland	Youngsville
Kenton	0	20	50	30
Akronia	20	0	45	100
Leveland	50	45	0	60
Youngsville	30	100	70	0