(CS60058) FAULT TOLERANT SYSTEMS

ANALYTICAL ASSIGNMENT-4

Submission Guidelines : Solve the following numerical problems in pen and paper. In your submission clearly mention your name and roll number.

1. Consider the following programs as two different versions of the same software.

```
F1(int x, int y)
{
 int a =10,b=5,z;
 if(x>a && y<a)
 {
  if(y<b) z= <0.01> f1(x,y);
  else    z= <0.02> f2(x,y);
 }
 elseif(y>25)
  z= <0.001> f3(x,y);
 else x= <0.003> f4(x,y);
 x= <0.01> f5(z);
 return x;
}
```

```
F1'(int x, int y)
{
 int a =10,b=5,z;
 if(x>a && y<a)
  z= <0.005> f1'(x,y);
 elseif(y>25)
 {
  if(x<=a) z= <0.001> f2'(x,y);
  else     z= <0.003> f3'(x,y);
 }
 else x= <0.003> f4(x,y);
 x= <0.01> f5(z);
 return x;
}
```

The quantities inside "<>" are the failure probabilities of the different function calls. The difference between the versions can be motivated as follows. Inside the region "$x > a \,\&\&\, y < a$", the first version uses two different functions (f1 and f2) inside two different subspaces. The functions perform the same job but it just happens that f1 has better numerical stability inside $y < b$ while f2 is better inside the other subspace. This motivates their choices. In the other version, we have f1' (an equivalent function) which is far more reliable (but slow due to serial re-execution). Similarly, f3, f2' and f3' are functionally same but have different reliabilities. Consider the probability distributions of inputs x and y as follows.

Variable	Interval	Probability
x	[0,10]	uniformly distributed with probabilty 0.7
	[10,40]	uniformly distributed with probabilty 0.3
	>40	uniformly distributed with probabilty 0.0
y	[0,5]	uniformly distributed with probabilty 0.25
	[5,10]	uniformly distributed with probabilty 0.6
	[10,40]	uniformly distributed with probabilty 0.15
	>40	uniformly distributed with probabilty 0.0

Compute failure probability of each individual version as well as the joint failure probability. **[3+3+2]**

2. Discuss the DFS based routing algorithm for injured Hypercubes. Elaborate on i) How does the algorithm select a detour when all optimal paths are blocked ? ii) What is the usefulness of backtracking in the algorithm? If the Hamming distance between two connected nodes is k, what is the no. of hops required to send a message in the worst case ? **[3+3+3+3]**