|  |  |  |
| --- | --- | --- |
|  | Cairo University | A picture containing diagram  Description automatically generated |
| Faculty of Computers and Artificial Intelligence |
| Software Engineering Program |
| **Algorithms Design and Analysis**  **Lecture Task 02** |

|  |  |  |  |
| --- | --- | --- | --- |
| Name: | Khalid Ibrahim Abdallah Shawki | ID: | 20206018 |

**Q1: A primality test is an algorithm for determining whether an input number is prime or not, present 4 different methods**

A primality test is an algorithm for determining whether an input number is prime. Among other fields of mathematics, it is used for cryptography.

4 Methods to Solve primality test:

1. School Method
2. Fermat Method
3. Miller-Rabin Method
4. Solovay Strassen Method

1. School Method

We Iterate through all number from 2 to n-1 and for every number check if it divides n, If we find any number that divides we return false. The complexity is O (n).

2. Fermat Method

1. Repeat following k times:
   1. Pick a randomly in the range [2, n - 2]
   2. If gcd(a, n) ≠ 1, then return false
   3. If an-1 &nequiv; 1 (mod n), then return false
2. Return true [probably prime].

3. Miller-robin Method

It returns false if n is composite and returns true if n is probably prime. k is an input parameter that determines accuracy level. Higher value of k indicates more accuracy.

bool isPrime(int n, int k)

1) Handle base cases for n < 3

2) If n is even, return false.

3) Find an odd number d such that n-1 can be written as d\*2r.

Note that since n is odd, (n-1) must be even and r must be

greater than 0.

4) Do following k times

if (millerTest(n, d) == false)

return false

5) Return true.

This function is called for all k trials. It returns false if n is composite and returns true if n is probably prime.

d is an odd number such that d\*2r = n-1 for some r>=1

bool millerTest(int n, int d)

1) Pick a random number 'a' in range [2, n-2]

2) Compute: x = pow(a, d) % n

3) If x == 1 or x == n-1, return true.

Below loop mainly runs 'r-1' times.

4) Do following while d doesn't become n-1.

a) x = (x\*x) % n.

b) If (x == 1) return false.

c) If (x == n-1) return true.

4. Solovay–Strassen Method

The Solovay–Strassen primality test is a probabilistic test to determine if a number is composite or probably prime. Before diving into the code we will need to understand some key terms and concepts to be able to code this algorithm.

Text

Description automatically generated

**Q2: in knapsack problem discuss how the backtracking algorithm work?**

Text, letter

Description automatically generated

Text, letter

Description automatically generated

**3. show the stpes of solving money change problem by dynamic programming**

Example: coins = {1, 2, 5}

Amount = 11

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 0 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 |

1) Min (0+1, 12) = 1

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 0 | 1 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 |

2) Min (1+1, 12) = 2

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 0 | 1 | 1 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 |

3) Min (1+1, 12) = 2

Min (1+1, 2) = 12

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 0 | 1 | 1 | 2 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 |

4) Min (2+1, 12) = 3

Min (1+1, 3) = 2

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 0 | 1 | 1 | 2 | 2 | 1 | 12 | 12 | 12 | 12 | 12 | 12 |

5) Min (2+1, 12) = 3

Min (2+1, 3) = 3

Min (0+1, 3) = 1

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 0 | 1 | 1 | 2 | 2 | 1 | 12 | 12 | 12 | 12 | 12 | 12 |

6) Min (1+1, 12) = 2

Min (2+1, 2) = 2

Min (1+1, 2) = 2

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 0 | 1 | 1 | 2 | 2 | 1 | 2 | 12 | 12 | 12 | 12 | 12 |

7) Min (2+1, 12) = 3

Min (1+1, 2) = 2

Min (1+1, 2) = 2

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 0 | 1 | 1 | 2 | 2 | 1 | 2 | 2 | 12 | 12 | 12 | 12 |

8) Min (2+1, 12) = 3

Min (2+1, 3) = 3

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 0 | 1 | 1 | 2 | 2 | 1 | 2 | 2 | 3 | 12 | 12 | 12 |

9) Min (3+1, 12) = 4

Min (2+1, 4) = 3

Min (2+1,3) = 3

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 0 | 1 | 1 | 2 | 2 | 1 | 2 | 2 | 3 | 3 | 12 | 12 |

10) Min (3+1, 12) = 4

Min (3+1, 4) = 3

Min (1+1, 3) = 3

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 0 | 1 | 1 | 2 | 2 | 1 | 2 | 2 | 3 | 3 | 2 | 12 |

11) Min (2+1, 12) = 3

Min (3+1, 3) = 3

Min (2+1, 3) = 3

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 0 | 1 | 1 | 2 | 2 | 1 | 2 | 2 | 3 | 3 | 2 | 3 |