Problem 1.1 solution

Given,

Graph G = (V, E) with

V = {a, b, c, d, e, f}

E={(a,b), (a,e), (a,f), (b,c), (b,f), (c,d), (c,f), (d,e), (d,f), (e,f)}

Now, lets construct a spanning tree G(V, E’) using Kruskal’s algorithm

|  |  |  |
| --- | --- | --- |
| **Start** | E = {}  A = {{a}, {b}, {c}, {d}, {e}, {f}} | C=0 |
| **Step 1:** | E = {(a, f)}  A = {{a, f}, {b}, {c}, {d}, {e}} | C=1 |
| **Step 2:** | E = {(a, f), (b, c)}  A = {{a, f}, {b, c}, {d}, {e}} | C=3 |
| **Step 3:** | E = {(a, f), (b, c), (d, e)}  A = {{a, f}, {b, c}, {d, e}} | C=6 |
| **Step 4:** | E = {(a, f), (b, c), (d, e), (d, f)}  A = {{a, f, d, e}, {b, c}} | C=10 |
| **Step 5:** | E = {(a, f), (b, c), (d, e), (d, f), (c, f)}  A = {{a, f, d, e, b, c}} | C=15 |
| **End** | E = {(a, f), (b, c), (d, e), (d, f), (c, f)}  V = {a, b, c, d, e, f} | C=15 |

Hence, the minimum value of the cost function is 15.

Problem 1.2 Solution

Given:

∑ = {L, R, F, P}

t = FFLFLFRFRFFLFRF

n = 15 (length of text t)

p = FFLFR

m = 5 (length of pattern p)

1.2a solution)

Using naïve string search algorithm,

FFLFLFRFRFFLFRF

FFLFR

FFlfr

Fflfr

FFlfr

Fflfr

FFlfr

Fflfr

FFlfr

Fflfr

FFLFR

No. of alignments = 10

No. of comparisons = 22

1.2a solution)

Using Boyer-Moore bad character rule algorithm,

FFLFLFRFRFFLFRF

fflfR

FFLFR

fflfR

ffLFR

fflfR

FFLFR

No. of alignments = 6

No. of comparisons = 16

1.2c solution)

Positions of Pattern p FFLFR

01234

No. of alignments that can be skipped if comparison does not match.

“-” means Comparison matches

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Post.**  **∑** | **F**  **0** | **F**  **1** | **L**  **2** | **F**  **3** | **R**  **4** |
| **F** | - | - | 0 | - | 0 |
| **L** | 0 | 1 | - | 0 | 1 |
| **R** | 0 | 1 | 2 | 3 | - |
| **P** | 0 | 1 | 2 | 3 | 4 |

The values in the table denote the number of alignments that can be skipped if there is a mismatch. Hyphen means the comparisons match.

Problem 1.3 solution

1.3a Answer

The operators which are neither right associative nor left associative are called non-associative operators. If such operators having same precedence level appear multiple times in an expression, then we will get a **precedence error.** The non-associative operators in Haskell are ==, /=, <, <=, >=, > which have precedence level 4.

e.g.

In [1]: 3 == 5 <= 4 /= 5

Cannot decide which operator to execute first

because they are non-associative and have

same precedence level.

1.3b Answer)

The precedence of $ is zero in Haskell and it is **right associative.**

The infix notation of (^) 2 $ (\*) 5 $ (+) 2 3 is 2^(5\*(2+3)).