# Time complexity Analysis

**Algorithm:**

**COST**

**Maximum M\_SUBJECT 8**  θ(1)

**Structure SUBJECT** θ(1)

{

SUBNAME

CHRS

SUBTYPE

}

**Structure TIMETABLE**  θ(1)

{

T\_ID

LEC1

LEC2

LEC3

LEC4

TBREAK

LEC5

LEC6

LEC7

}

**SLOT\_ALLOCATION (SUBJECT S[ ], TIMETABLE T[ ] , M\_SUBJECT)**  θ( n )

{

Let WEEKDAY = 1 θ(1)

For i=1 to M\_SUBJECT θ( n )

{

If ( i=1 | | i =3 || i =5 || i =7 ) θ( n - 1)

{

if (S[ i ].SUBTYPE == "Lab" || S[ i ].SUBTYPE == "lab" || S[ i ].SUBTYPE == "l") θ( n - 1)

{

T[ WEEKDAY ].LEC1=S[ i ].SUBNAME θ( n - 1)

T[ WEEKDAY ].LEC2=S[ i ].SUBNAME θ( n - 1)

T[ WEEKDAY ].LEC3=S[ i ].SUBNAME θ( n - 1)

}

else if(S[ i ].SUBTYPE =="Theory" || S[ i ].SUBTYPE == "theory" || S[ i ].SUBTYPE == "th")

θ( n - 1)

{

if(S[ i ].CHRS=="1") θ( n - 1)

{

T[ WEEKDAY ].LEC1=S[ i ].SUBNAME θ( n - 1)

T[ WEEKDAY ].LEC2= “ – ” θ( n - 1)

T[ WEEKDAY ].LEC3= “ – ” θ( n - 1)

}

else if(S[ i ].CHRS =="2") θ( n - 1)

{

T[ WEEKDAY ].LEC1=S[ i ].SUBNAME θ( n - 1)

T[ WEEKDAY ].LEC2=S[ i ].SUBNAME θ( n - 1)

T[ WEEKDAY ].LEC3= “ – ” θ( n - 1)

}

else if(S[ i ].CHRS =="3") θ( n - 1)

{

T[ WEEKDAY ].LEC1=S[ i ].SUBNAME θ( n - 1)

T[ WEEKDAY ].LEC2=S[ i ].SUBNAME θ( n - 1)

T[ WEEKDAY ].LEC3=S[ i ].SUBNAME θ( n - 1)

}

}

Else θ( n - 1)

{

T[ WEEKDAY ].LEC1= “ – ” θ( n - 1)

T[ WEEKDAY ].LEC2= “ – ” θ( n - 1)

T[ WEEKDAY ].LEC3= “ – ” θ( n - 1)

}

}

If ( i =2 || i =4 || i =6 || i =8) θ( n - 1)

{

if (S[ i ].SUBTYPE == "Lab" || S[ i ].SUBTYPE == "lab" || S[ i ].SUBTYPE == "l") θ( n - 1)

{

T[ WEEKDAY ].LEC4= “ – ” θ( n - 1)

T[ WEEKDAY ].TBREAK = “BREAK” θ( n - 1)

T[ WEEKDAY ].LEC5=S[ i ].SUBNAME θ( n - 1)

T[ WEEKDAY ].LEC6=S[ i ].SUBNAME θ( n - 1)

T[ WEEKDAY ].LEC7=S[ i ].SUBNAME θ( n - 1)

}

else if((S[ i ].SUBTYPE == "Theory" || S[ i ].SUBTYPE == "theory" || S[ i ].SUBTYPE == "th") &&

(S[ i – 1 ].SUBTYPE == "Theory" || S[ i – 1 ].SUBTYPE == "theory" || S[ i – 1 ].SUBTYPE == "th") &&

(S[ i - 1 ].CHRS || S[ i – 1 ].CHRS == "2" || S[ i – 1 ].CHRS == "3")) θ( n - 1)

{

if (S[ i ].CHRS == "1") θ( n - 1)

{

T[ WEEKDAY ].LEC4=S[ i ].SUBNAME θ( n - 1)

T[ WEEKDAY ].TBREAK = “BREAK” θ( n - 1)

T[ WEEKDAY ].LEC5= “ – ” θ( n - 1)

T[ WEEKDAY ].LEC6= “ – ” θ( n - 1)

T[ WEEKDAY ].LEC7= “ – ” θ( n - 1)

}

else if (S[ i ].CHRS == "2") θ( n - 1)

{

T[ WEEKDAY ].LEC4=S[ i ].SUBNAME θ( n - 1)

T[ WEEKDAY ].TBREAK = “BREAK” θ( n - 1)

T[ WEEKDAY ].LEC5= S[ i ].SUBNAME θ( n - 1)

T[ WEEKDAY ].LEC6= “ – ” θ( n - 1)

T[ WEEKDAY ].LEC7= “ – ” θ( n - 1)

}

else if (S[ i ].CHRS == "3") θ( n - 1)

{

T[ WEEKDAY ].LEC4=S[ i ].SUBNAME θ( n - 1)

T[ WEEKDAY ].TBREAK = “BREAK” θ( n - 1)

T[ WEEKDAY ].LEC5= S[ i ].SUBNAME θ( n - 1)

T[ WEEKDAY ].LEC6= S[ i ].SUBNAME θ( n - 1)

T[ WEEKDAY ].LEC7= “ – ” θ( n - 1)

}

}

else if ((S[ i ].SUBTYPE == "Theory" || S[ i ].SUBTYPE == "theory" || S[ i ].SUBTYPE == "th") &&

(S[ i - 1 ].SUBTYPE == "Lab" || S[ i – 1 ].SUBTYPE == "lab" || S[ i – 1 ].SUBTYPE == "l")) θ( n - 1)

{

if (S[ i ].CHRS == "1") θ( n - 1)

{

T[ WEEKDAY ].LEC4=S[ i ].SUBNAME θ( n - 1)

T[ WEEKDAY ].TBREAK = “BREAK” θ( n - 1)

T[ WEEKDAY ].LEC5= “ – ” θ( n - 1)

T[ WEEKDAY ].LEC6= “ – ” θ( n - 1)

T[ WEEKDAY ].LEC7= “ – ” θ( n - 1)

}

else if (S[ i ].CHRS == "2") θ( n - 1)

{

T[ WEEKDAY ].LEC4=S[ i ].SUBNAME θ( n - 1)

T[ WEEKDAY ].TBREAK = “BREAK” θ( n - 1)

T[ WEEKDAY ].LEC5= S[ i ].SUBNAME θ( n - 1)

T[ WEEKDAY ].LEC6= “ – ” θ( n - 1)

T[ WEEKDAY ].LEC7= “ – ” θ( n - 1)

}

else if (S[ i ].CHRS == "3") θ( n - 1)

{

T[ WEEKDAY ].LEC4=S[ i ].SUBNAME θ( n - 1)

T[ WEEKDAY ].TBREAK = “BREAK” θ( n - 1)

T[ WEEKDAY ].LEC5= S[ i ].SUBNAME θ( n - 1)

T[ WEEKDAY ].LEC6= S[ i ].SUBNAME θ( n - 1)

T[ WEEKDAY ].LEC7= “ – ” θ( n - 1)

}

}

Else θ( n - 1)

{

T[ WEEKDAY ].LEC4=“ – ” θ( n - 1)

T[ WEEKDAY ].TBREAK = “BREAK” θ( n - 1)

T[ WEEKDAY ].LEC5= “ – ” θ( n - 1)

T[ WEEKDAY ].LEC6= “ – ” θ( n - 1)

T[ WEEKDAY ].LEC7= “ – ” θ( n - 1)

}

WEEKDAY = WEEKDAY+ 1 θ( n - 1)

}

}

}

Moreover, fitness is the prime function and it decides which will be selected and it takes maximum time, which is crucial for the whole algorithm and our algorithm it takes **Ѳ(n5).** A genetic algorithm is a polynomial time algorithm. We are trying to improve the time complexity of our algorithm and we will work to optimize our algorithm to reduce its time complexity.

**Correctness of algorithm**

**Introduction:**

**Correctness of algorithm**

**Conclusion:**