**Master’s Theorem: T(n) = aT(n/b)+f(n)** Where f(n)= n^d

Time complexity problems:

1. T (n) = 3T (n/2) + n

According to Master’s theorem

a= 3, b=2, f(n)=n therefore d = 1

here a>b^d

therefore case 3 applies T(n) = 🞊 (n^)

Therefore, substitute the values we get

**T(n) = 🞊(n^)**

1. T (n) = 64T (n/8) − n^2(log n)

**Here f(n) is negative, Master’s theorem doesn’t apply**

1. T (n) = 2nT (n/2) + n^n

a = 2n,

b = 2,

f(n) = n^n, d=n

a<b^d therefore case 1 applies T(n) = 🞊(n^d)

where n>2

**T(n) = 🞊(n^n)**

{if n=1,2 then case 2 of master’s theorem will apply since a=b^d

T(n) = 🞊(n^d log n); T(n) =0 for n=1

T(n) = 🞊(n^d log n); T(n) = (4log2) for n = 2 }

1. T (n) = 3T (n/3) + n/2

a = 3,

b = 3

f(n) = n/2, d = 1

a = b^d therefore case 2 applies T(n) = 🞊(n^d log n)

**T(n) = 🞊(n log n)**

1. T (n) = 7T (n/3) + n^2

a = 7,

b = 3

f(n) = n^2, d = 2

a < b^d therefore case 1 T(n) = 🞊(n^d)

**T(n) = 🞊(n^2)**