Binary Search Algorithm

Binary Search is an efficient algorithm for performing a search for a target value T on an ordered (non-decreasing) list of length n. The algorithm employs a divide and conquer method as follows. We start with L=0 and R=n-1 and perform the following.

1. Set m=floor[(L+R)/2].
2. Compare target value to mth value in list, if target value is equal, then we are done; if target value is less, then set R=m-1; otherwise set L=m+1.
3. Repeat...

Note that the best case scenario takes 1 comparison, and the worst case scenario requires $Log_2[n]$ comparisons. Let (S,P) denote the sample space of all possible searches, assuming that all values are equally likely to be searched for. Denote by X(s) the random variable which represents the number of iterations needed to perform the search s. The following derives the expectation value, variance, and standard deviation of the random variable X.

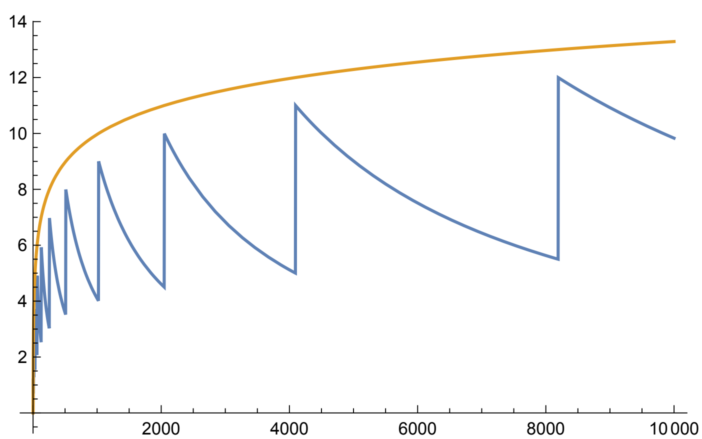
Expectation Value

In[10]:=

```
F[n_]:=Sum[k*(2^(k-1)/n),{k,1,Log2[n]}]
Simplify[F[n]]
Plot[{F[n],Log2[n]},{n,1,10000}]
```

Out[11]= $-1+\frac{1}{n}+\frac{\text{Log}[n]}{\text{Log}[2]}$

Out[12]=

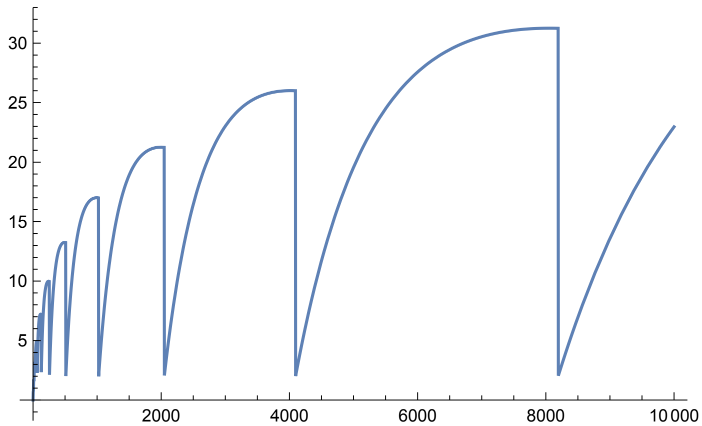


Variance

In[13]:=
```
V[n_]:=Sum[k^2*(2^(k-1)/n),{k,1,Log2[n]}]-F[n]^2
Simplify[V[n]]
Plot[V[n],{n,1,10000}]
```

Out[14]= $-\frac{\text{Log}[2]+n\,\text{Log}[2]-n^2\,\text{Log}[4]+2\,n\,\text{Log}[n]}{n^2\,\text{Log}[2]}$

Out[15]=



Standard Deviation

In[16]:=
```
S[n_]:=Sqrt[V[n]]
Simplify[S[n]]
Plot[S[n],{n,1,10000}]
```

Out[17]= $\frac{\sqrt{-\frac{\text{Log}[2]+n\,\text{Log}[2]-n^2\,\text{Log}[4]+2\,n\,\text{Log}[n]}{n^2}}}{\sqrt{\text{Log}[2]}}$

Out[18]=