```
likelyWeight<-function(alpha,zValue,bitsArr){
  fb <- 0

  for (index in 1:length(bitsArr)){
    fb <- fb + bitsArr[index] * (2 ** (index - 1))
  }

  prob = (1-alpha)/(1+alpha)*(alpha ** abs(zValue - fb))
  return (prob)
}

approInfer<-function(samplesize, bitNum, i, alpha, Zvalue){
  gene <- c(0,1)
  numerator <- 0
  denominator <- 0

  for (count in 1:samplesize){
    bitSet <- integer(bitNum)

    for(index in 1:bitNum){
      bitSet[index]<-sample(gene,1,replace=TRUE)
    }

    pro = likelyWeight(alpha, Zvalue, bitSet)
    denominator <- denominator + pro

    if(bitSet[i] == 1){
      numerator <- numerator + pro
    }
  }

  return (numerator/denominator)
}



sampleNum <- seq(10000,500000,by=10000)
probValueSet <- integer(50)

for (currsize in 0:50){
  probValueSet[currsize] <- approInfer(currsize * 10000, 10, 10, 0.1, 128)
  print(currsize * 10000)
}

plot(sampleNum, probValueSet,type='l')
```