1 SVD implementation

```
# Calculate probabilities p_i and use it as probibility distribution for constructing matrix S
n,m = A.shape
p = np.zeros(n)
fro = np.linalg.norm(A[1:], ord = ’fro’) ** 2
for i in range(1, n):
    p[i] = (np.linalg.norm(A[i])**2)/fro
# Construct S matrix of size s by m using p prob distribution
S = np.zeros((s,m))
for i in range(1,s):
    j = np.random.choice(n, p = p, replace=False)
    S[i] = A[j]
# Calculate SS^T
sst = np.matmul(S, S.T)

# compute SVD for SS^T
u, s , vh = np.linalg.svd(sst)
lamb = np.sqrt(s[:k])

# Construct H matrix of size m by k
H = np.zeros((k,m))
for i in range(1,k):
    st_w = np.matmul(S.T, vh[i])
    H[i] = st_w/np.linalg.norm(st_w)
H = H.T
print("shape of H: ", H.shape)

# Return matrix H and top-k singular values sigma
return H, lamb
```