**int** len = A.length;

**if**(len == 0 || len < D) **return** 0;

**int**[] re = **new** **int**[len+1];

**int** [] arriveTime = **new** **int**[len+1];

**for**(**int** i =0; i< len; i++) arriveTime[i] = A[i];

Arrays.*fill*(re, 100001);

**for**(**int** pos =0; pos<=len; pos++){

**if**(pos+1<=D) re[pos] = arriveTime[pos];

**else**{

**for**(**int** step = 1; step<=D; step++){

**if**(arriveTime[pos]>re[pos-step]){

re[pos] = Math.*min*(re[pos], arriveTime[pos]);

}

**else**{

re[pos] = Math.*min*(re[pos], re[pos-step]);

}

}

}

}

**return** re[len] == 100001?-1:re[len];

If length of A is 0 , which meas that we do not need the stone.

arriveTime array is a copy of array A, just add 0 to the end, which meas that at detination N, it has stone at the time zero. Which is prepared for our later processing.

Also use a dp array, named result to store earlies time that we could reach the specific spot, the spot index is responding to the result array index.

Then we use dynamic programming method, loop the index of the stone from 0 to N , to find their ealiest reachment time.

Pos represent the index of spot.

If pos + 1 smaller than d which means that we could reach the destination N in time 0, we do not need the stone.

If we could reach the pos in one step, we need to find if we could reach the pos by utilizing the stone , the pos of which is smaller than the one now.

 I heard that the colleagues in the liveramp arr all incredibly talented: super competent at their jobs, very thoughtful, and super hungry to grow and improve themselves. (B) They're really, really nice.

There's so much opportunity to grow not just within your role at the company,

We detail with enormous amounts of data, and have had to reinvent the way we process it a number of times to deal scale. People have a ton of responsibility, and each individual can have an enormous impact on the success and trajectory of the company.