**void simplemerge(char \*\* a, int low, int mid, int high){**

**int i;**

**int j;**

**int k;**

**char \*\*c=malloc((high-(low-1))\*sizeof(char\*));**

**i=low;**

**j=mid+1;**

**k=0;**

**while(i<=mid && j<=high){**

**if((strlen(a[i])>strlen(a[j])) || ((strlen(a[j])==strlen(a[i])) && (strcmp(a[j],a[i])<0))){**

**c[k]=a[i];**

**k=k+1;**

**i=i+1;**

**}**

**else{**

**c[k]=a[j];**

**k=k+1;**

**j=j+1;**

**}**

**}**

**while(j<=high){**

**c[k]=a[j];**

**j=j+1;**

**k=k+1;**

**}**

**while(i<=mid){**

**c[k]=a[i];**

**k=k+1;**

**i=i+1;**

**}**

**for(i=0;i<high-low+1;i++){**

**a[i+low]=c[i];**

**}**

**}**

**void kthlargestnumber(char \*\* a, int low, int high){**

**if(low<high){**

**int mid;**

**mid=(low+high)/2;**

**kthlargestnumber(a, low, mid);**

**kthlargestnumber(a, mid+1, high);**

**simplemerge(a,low,mid,high);**

**}**

**}**

**char\* kthLargestNumber(char\*\* nums, int numsSize, int k) {**

**kthlargestnumber(nums, 0, numsSize-1);**

**return(nums[k-1]);**

**}**

