Tutorial - 3 int Lunear - Servich (int or avon., int n, ent key)

for i >0 (to n-1

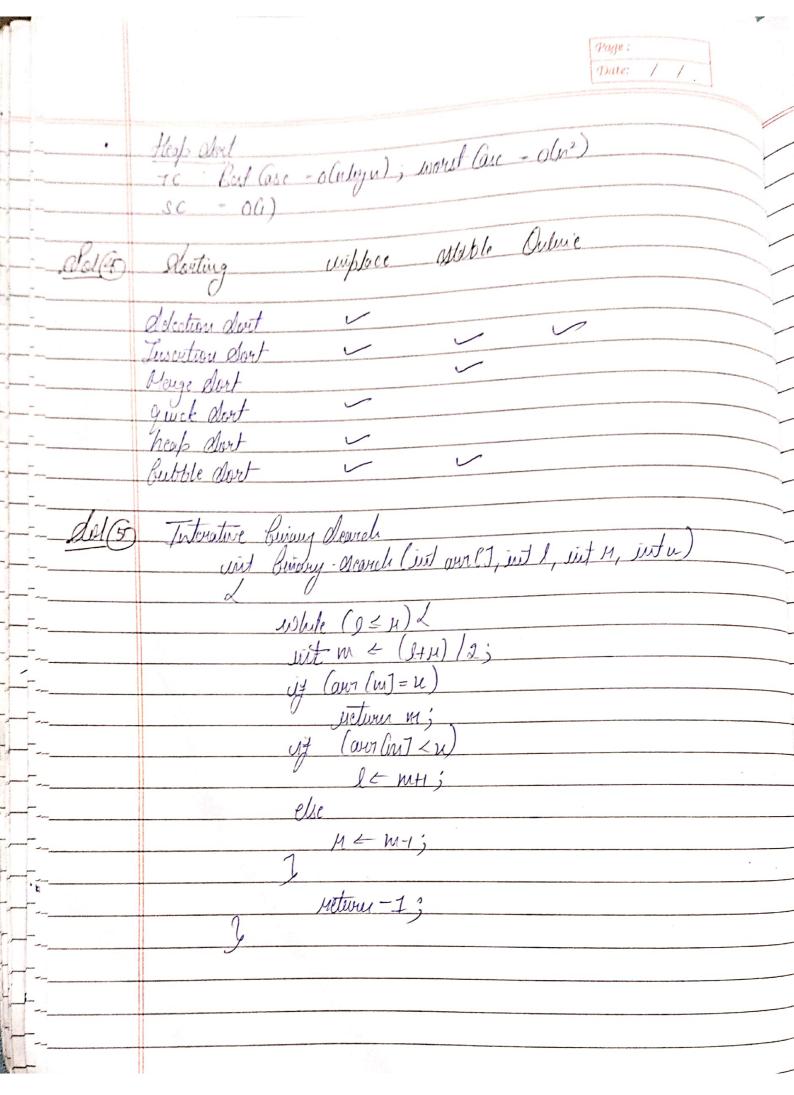
if avon G J = key

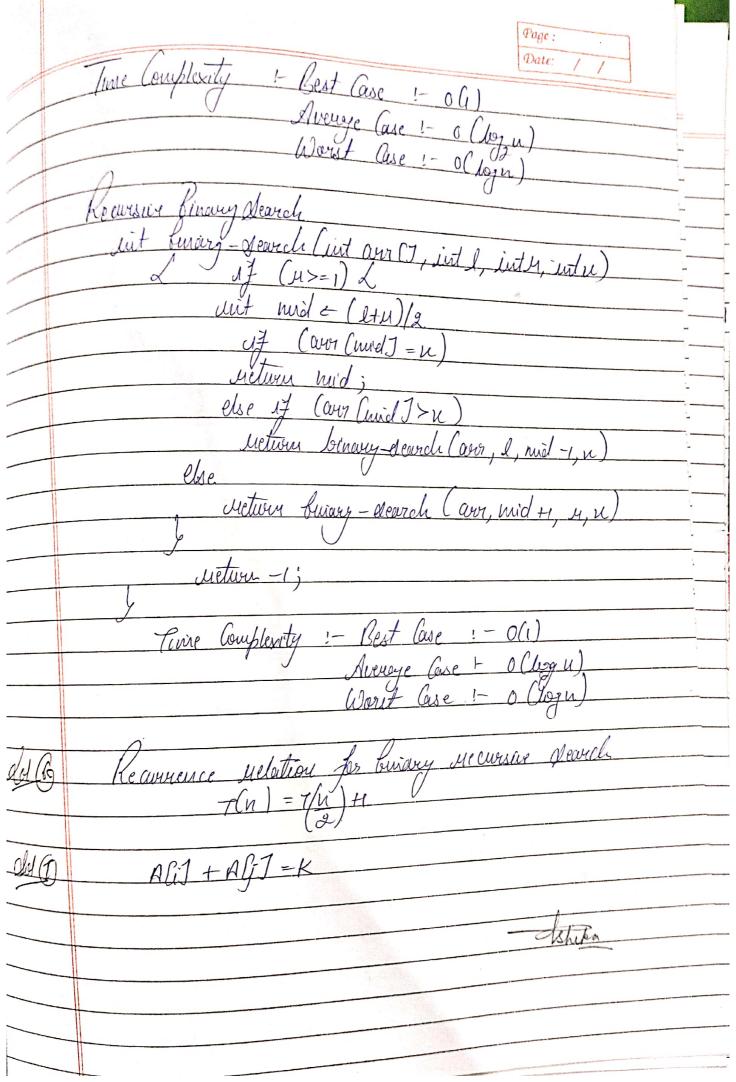
Justinum i Melion -1 Moundaix unscriber doct 120 Void ensortion-dood (wit over (1, wit u) deup < any aj while (g>=0 AND OUT GJ > Temp)

OUT GHT - OUT GJ Me cursur insertion Void inscrition-sout (int aux C), int u insertion port Carr n-1 Shika EJS Isheka DAA

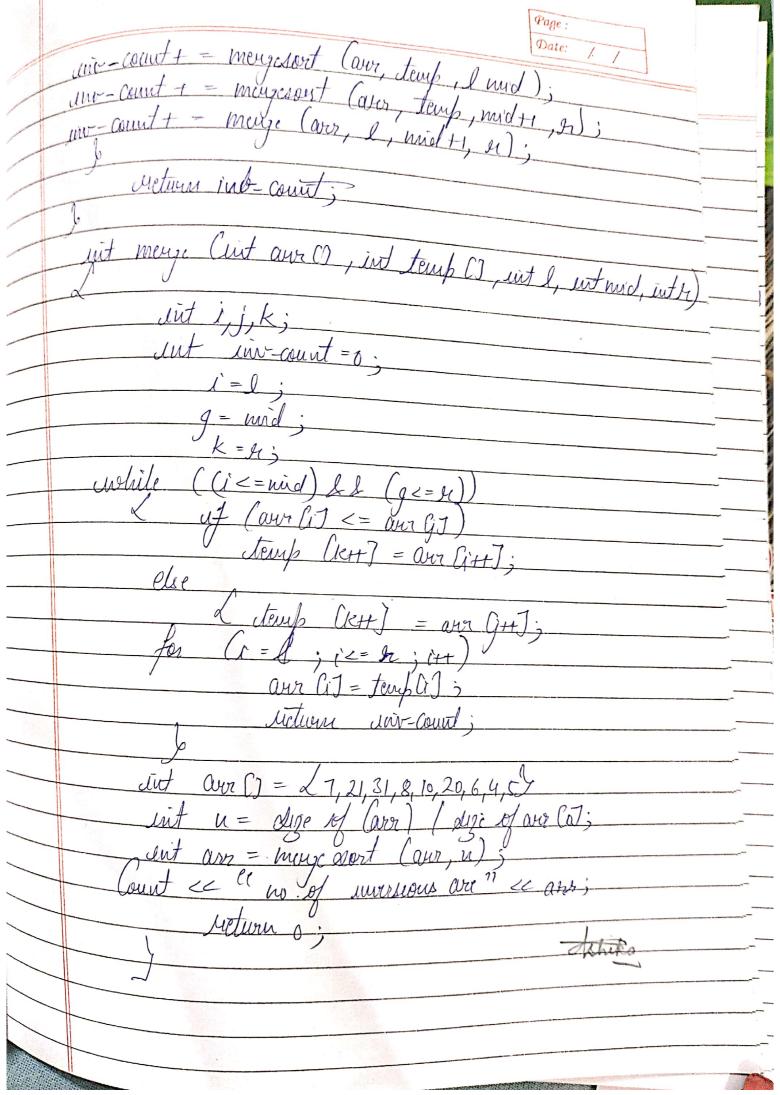
	Page:	
	while (j>=0 Is any G1> 1.1)	/ /
	while (j>=0 del ann GJ > last) ann GHJ = ann GJ	
- Control of the Cont	4	
The state of the s	aug (j+1) = Last	
	aux (j+1) = Last airention sout its realled online souting because it need to know anything about what values it sout and the Information is required while algorithm is running	1 -4
	need to know anything about what values it	will will
	algorithus is running is required while	tue
	agorifor in ranning	
1/1/20	· Delection Sout	
MO	Time Complexity: best case: - 0 (n2); werst case	
	Space Complexity: 0(1)	= 0 (h²)
	· Inscrition Sout	
	TC: best Case: O(u); worst case=0(u2)	
	s.c :- 00)	
	a 11. Aut	
	Menze Chart To = Bost Constant = a Contract Constant Con	
	7.c = Best Complexity :- 0 (nleg n); worst Case S.c = O(n)	=0(N/194)
	· Quick dort	
	T.C : Best Case 1-0 (nlogn); worst Case = $0(n^2)$ S.C :- $0(n)$)
	S-C :- O(n)	
	O = O = O = O	
	Bubble dort	
	7.c - Best Case = Co (n²), Worst Case = o(n²)	
	S.C = O(1)	

Scanned by CamScanner





Date: Ruck dort in the failest general purpose dort, In most farctical stitution gurck sout is the method of choice of klability is important and apace is circubeble, energy wort, myld be best: Inversion count for any away inchreates :- how far always is from feing souted If the array us always is orted, they the inverseous court is 0, but if away is souted air the never e order, the inverseous - avr[] = {7,21,31,8,10,1,20,6,4,5} # include < bits (etdc ++ h) Mying mannespace setol; suit - mange soot (int and [] int temp [], int l, int en. dut maye don't (int aun 1), int away size Lite temp (armany-size); Netwur - prengesiert (ans, Temp, o, annoy-size list und, inv-court =0: mid = (4+1)/2;



	Page
	Date / /
	ut of auckwart is 6(n2)
asoller	The worst case time complexity of quick wort is one
- Tion	The worst case occur when the picked from hoffen
The state of the s	
	on extreme (smallest or largest) element, this propper
	alon dubut away to myca or
The second secon	With the start of left (TELMON)
Specialistic design and convenience in the convenience of the convenie	The best case of Quick Ant is select see sell Added first as a mean element
	the best Case of Chick Mart is process
	delect biret as a mean element
-	
- Notes	
- MARCO	Keaurence relation of:
	a) Meye Gort: - 7(n) = 27 (1/2) + n b) Quek dort: - 7(n) = 27 (n/2) + n
	b) Quel dot (- 76.) = 07 (10/2) 11.
	- Clark My 1 (N1 - 21 (192) + N
7	Maye glort is more efficient and works getter Alice
	quick lost in case of lace and the
	guick Nort in case of large array size of dataset
	Worst Case Complainty for Cluick and is 0/42) whereas 6 Culog of for many dort
~	dishereas 6 Culos VI for mouse dost
	1 - waste with
-	
~	
	chile.
~	
16	
~	
-	

Atath delection dort Stath delection dort Stath delection dort Stath delection dort (int a G, int n) for (int i=0; i2n-1; i1+) wint nun-1; for (int g=1+1; j <n; (a="" (inin="" 1="" j++)="" yf=""> a G, i) min = z; wint key = a (inid]; wintile (min z) Lumile (min z) y a G = key; y wint nain () L wint o G = (45,3,1,4/2; wint n = min of (a, u); for (int i=0; i<n; a="" con;="" condition="" condition<="" cont="" g,="" int)="" of="" section="" th="" the="" ~=""><th></th><th></th></n;></n;>		
Statle Selection dort Statle Selection dort Statle Selection sout (int a [], int n) Love with 1=0; (< N-); 1++) Cut num -1; for (int g-1+1; j< n; j++) The conting of the selection of the		Page:
Mathe Relection Root (int a G), int n) Lord stable detection sort (int a G), int n) Lord stable detection sort (int a G), int n) Lord stable detection sort (int a G), int n) Lord stable detection sort (int a G), int n) Lord stable detection sort (int a G), Lord stable (int a G); Lord stable detection dort (a, n); Lord stable detection dort (a,		Date:
Illing namesface sold; Vocal stable detection sout (int a s), int n for (int j=0; i< n-1; i+t) unt nuis -1; for (ut g=1+1; j< n; j+t) y (a anis) > a s) min = y; unt key = a loud); untile (nuis >1) La a lound = a lound -1; hain; y a s] = key; y uit unois () L uit a s] = L 45,3,2,4,1; uit n = mye of (a); atatle detection dont (a, u); for (nit s=0; i< n; i+t) aut ~ a s] ~ (i n; aut ~	Nexte delection dort	
Jer (wit 9 = 1+1; j × n; j++) Jer (wit 9 = 1+1; j × n; j++) Jer (a Guin] > a G J) Muin = 7; Muin Key = a Guid J; Mushile (num > 1) Land (num > 1) La	Maria Munestone with	distance and
Jer (wit 9 = 1+1; j × n; j++) Jer (wit 9 = 1+1; j × n; j++) Jer (a Guin] > a G J) Muin = 7; Muin Key = a Guid J; Mushile (num > 1) Land (num > 1) La	The stable of the state	
Jan (wit 9 = 1 + 1 ; j × n ; j + t) Jan (wit 9 = 1 + 1 ; j × n ; j + t) Jan (wit 1 > a G J) Muni	Void de de clection stort (int a C)	the control of the co
Jer (wit 9 = 1+1; j × n; j++) Jer (wit 9 = 1+1; j × n; j++) Jer (a Guin] > a G J) Muin = 7; Muin Key = a Guid J; Mushile (num > 1) Land (num > 1) La	α for (int $i=0$) (α)	yn)
Jer (wit 9 = 1+1; j × n; j++) Jer (wit 9 = 1+1; j × n; j++) Jer (a Guin] > a G J) Muin = 7; Muin Key = a Guid J; Mushile (num > 1) Land (num > 1) La		The state of the s
## (& Guin] > a G] Min = 7; With key = a Guid]; With ite (win > 1); Min; Just main () L Wit a G] = L 4,5,3,2,4,5; Wit under a Good of the continuation of the	ant mun =1:	
muin = 3; unit key = a (mid); ushile (min > 1); de a (min - 1); huin; y a (i) = key; y unit a (j = 1 4, 5, 3, 2, 4, 1); unit a = mine of (a); Atatle acception dont (a, u); for (int i = 0; i < u; i+t) Gent <= a (i) < c ? " Cout <= a (i) <= a (i) < c ? Cout <= a (i) <= a (i) < c ? Cout <= a (i) <= a (i) < c ? Cout <= a (i) <= a (i) < c ? Cout <= a (i) <= a (i) < c ? Cout <= a (i) <= a (i) < c ? Cout <= a (i) <= a (i) < c ? Cout <= a (i) <= a (i) < c ? Cout <= a (i) <= a (i) < c ? Cout <= a (i) <= a (i) < c ? Cout <= a (i) <= a (i) < c ? Cout <= a (i) <= a (i) < c ? Cout <= a (i) <= a (i) < c ? Cout <= a (i) <= a (i) <= a (i) < c ? Cout <= a (i) <=	to fuit o	And the second s
Musin = 3; Wishile (min > 1); Wishile (min > 1); Musin; Musin; Musin; Musin a (] = L 4, 5, 3, 2, 4, 1 ; Wit a = Mine of (a); Mtatle Aclection Wort (a, u); For (mit != 0; ! < u; !+t) Gut <= a (i] <= c !!	IN COC TILLY	A STATE OF THE PROPERTY OF THE
2 a Cuiv J = a Cuin -1]; huin; y a Gi J = key; y uit main () L uit a G J = L 45,3,2,4,4,7; vit n = me of (a); Atable delection dont (a, u); for (int i = 0; i < u; i+t) Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca ci i Cout < ca Ci J < ca ci i Cout		
2 a Cuiv J = a Cuin -1]; huin; y a Gi J = key; y uit main () L uit a G J = L 45,3,2,4,4,7; vit n = me of (a); Atable delection dont (a, u); for (int i = 0; i < u; i+t) Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca ci i Cout < ca Ci J < ca ci i Cout	min = 3	
2 a Cuiv J = a Cuin -1]; huin; y a Gi J = key; y uit main () L uit a G J = L 45,3,2,4,4,7; vit n = me of (a); Atable delection dont (a, u); for (int i = 0; i < u; i+t) Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca ci i Cout < ca Ci J < ca ci i Cout	Unit key = a Chuid 1	
2 a Cuiv J = a Cuin -1]; huin; y a Gi J = key; y uit main () L uit a G J = L 45,3,2,4,4,7; vit n = me of (a); Atable delection dont (a, u); for (int i = 0; i < u; i+t) Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Gi J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca Ci J < ca ci i Cout < ca ci i Cout < ca Ci J < ca ci i Cout	undrite (min >1)	
Muin; $a[j] = key;$ Juit main () L wit $a[j] = L = L = L = L = L = L = L = L = L = $	d a fund	
Jack = key; Just main () L wif a CJ = L 453,2,4/2; wit n = mine of (a); Atable Accolor Nort (a, u); for (int 1 = 0; 1 < u; 1 tt) Cout < c a Ci J < c c n; Cout < c condi;	$\frac{uuv}{uv} = 0 \left[uu - 1 \right];$	
Just main () L unt a () = L 4,5,3,2,4/2; unt n = mine of (a); Atable Ackerting Nort (a, u); for (int 1=0; 1'< n; 1+1) Cont <= a (i) <= c n; Cont <= condi;	y nun;	
Just main () L unt a [] = L 4,5,3,2,4/2; unt n = more of (a); Atable Adection Nort (a, u); for (int 1=0; 1 < n; 1+1) Cout < a [i] < e n; Cout < e condi;	0.00	
unt $a (J = \begin{cases} 4,5,3,2,4,1 \end{cases}$; unt $n = M$ ye $xf(a)$; Atatle Acleban Nort (a, u) ; for $(xit) = 0$; $(x u) = 1$; Cout $(x u) = 1$. Cout $(x u) = 1$.	V = kcy	-
unt $a (J = \begin{cases} 4,5,3,2,4,1 \end{cases}$; unt $n = M$ ye $xf(a)$; Atatle Acleban Nort (a, u) ; for $(xit) = 0$; $(x u) = 1$; Cout $(x u) = 1$. Cout $(x u) = 1$.	Y J	0
unt $a (J = \{4,5,3,2,4,1\};$ unt $n = Mese \ tf(a);$ Atatle Acleban Nort $(a, u);$ for $(int \ l = 0; l < u; l + t)$ Cout (a, u) Cout (a, u)		
Just $n = Mese \ kf(a);$ Atatle Acleology Wort $(a, u);$ for $(sint \ i = 0; i < u; i + t)$ Cout $< < < < < < < < < < < < < < < < < < <$		
Atatle Acleolisu Nort (q, u); for (int 1 = 0 ; 1 < u ; 1+t) Cout << a (i) << e (1) Cout << e (nd);	1 9.5 2 9 (1/6)	
Atatle Aclection Nort (q, u); for (sit 1 = 0; 1 < u; 1++) Cout < c a (i) < c !! Cout < c endl;	Unt h = Mire of (a);	
Cout << end;		***************************************
Cont << end;	for (sit 1 = 0 ; 1 < 11 ; 1 ; 1)	
Cout << endl;		•
Metumn 0;		*
Shike	Metuna O:	4
- Shike	The state of the s	*
Shike		3
NINE O	Edit.	And the second s
	ishue of	1

	Page: Date: //
= 2000	The casest way to use extornal dorting use clivide our dource file auto tlemporary file of size equal ito dir if the Ram & first Stort these files
	External dosting! - If the input dater in such that it cannot ordinated in the manory enterry at once
	external dorting Luteyed dorting: If the infact data is such light
	it can be adjusted in the main memory at once— it is Called Internal Marting
'È	