Application of Heap PRIORITY QUEUES > Applications of a heap is paroarty queele > There are two kinds of pair only queue (1) Man- privarily general (3) Mois - Pacparty queens Set 'S' of elements, each with an associated value coulted A max- pararity queue supports the following operations. (1) INSERT (Six) - conserts the process elegant a costs the set S'. This operation is worther as St SUfag returns the element of S couth then or largerter becounted removes and returns the elements of cooth the longest key. (4) INCRESE-KEY (S, x, K) increases the value of element new value K .. e. Pocreses the of element & one apploration of max-partoraty quele is to schedule jobs on a shared computer on the to execute of procority finised ots opercuteron, the speak selected from those pending jobs is colded to the queene for INSERT. oshen a job is executing, at the some frome a higher a goves enterrupt highest porority job is elling EXTRACT-MAX. This highest queue for execution A porio - porio orty Sceppo 875 operate ons INSERT, MINIMUM EXTRACT-MIN, and beioscith used on an event quene

The procedure HEAP EXTRACT WAX oraplements the Mexicology of the Method of the Mexicology of the Mexic		3
I MENT MANDOM CAL I MENT PROCEDURE HEAP MANDOM Complements the Moncrowing operation, and action of the procedure HEAP MANDOM Complements the Moncrowing operation, and the procedure HEAP extract man complements the Extract man operation. The procedure HEAP Extract man complements the Extract man operation. The procedure HEAP Extract man conflements the Extract man operation. The procedure HEAP Extract man conflements the Extract man conflements the Extract man operation. The procedure the procedure of HEAP Extract man conflements the man HEAP INCREASE - ICEY (A in key) If key (A ii) The procedure HEAP INCREASE - KEY complements the PARENT(II) The procedure HEAP INCREASE - KEY complements the INCREASE - KEY operation. The procedure HEAP INCREASE - KEY complements the INCREASE - KEY operation. The procedure of HEAP INCREASE - KEY complements the INCREASE - KEY operation. The procedure of HEAP INCREASE - KEY complements the INCREASE - KEY operation. The procedure of HEAP INCREASE - KEY complements the INCREASE - KEY operation.	\rightarrow	To implement man paroanty quiene the D'operaterns are
The procedure HEAR IMANIMUM simplements the maximum operation of COLD devoc. (a) HEAR EXTRACT—MAX (A) 1. If heap-size[A] < 1 2. they exare heap emoler flow." 3. man (ACI) (heap-size[A]) 5. heap-size[A] (heap-size[A]) 6. MAX-HEAPIFY (AXI) 7. Yeturn man. The procedure HEAP-EXTRACT MAX complements the EXTRACT—MAX of O(1877), Decounted max-HEAPIFY toxics O(1977) The running from of HEAP-EXTRACT—MAX is O(1877), Decounted max-HEAPIFY toxics O(1977) 1. If key (ACI) 2. then error get key is smaller than current key. 3. ACI] (Rey) 4. colore ix 1 and ACPARENT(i)] (ACI) 5. do exchange ACI] (ACPARENT(i)] 6. ix PARENT(i) The procedure HEAP-INICREASE-KEY on plements the INICREASE-KEY operation. The auronag from at HEAP-INICREASE-KEY on on n-clement heap is O(1697) become the path traced from node is	(1)	HEAP_MAXIMUM CA)
The procedure HEAR MAXIMINA implements the maximum operation of DCI) devoc. (a) HEAR EXTRACT—MAX (A) 1. If heap-size[A] < 1 2. Then extor's heap emoler flow" 3. MAX HEAP EXTRACT—MAX (BY) 4. YEAR SIZE[A] & heap-size[A]] 5. Heap-size[A] & heap-size[A]] 5. Heap-size[A] & heap-size[A]] 6. MAX HEAP IFY (AXI) 7. YEAR TIMEN OPERATION The procedure HEAP EXTRACT MAX complements the EXTRACT—MAX of O(1879), Decount MAX—HEAPIFY forces O(1979) The running fine of HEAP-EXTRACT—MAX is O(1879), Decount MAX—HEAPIFY forces O(1979) The procedure HEAP INICE ASE—ICEY (A, is key) 1. If key (A[i] 2. then error's geod key is smaller than current key". 3. A[i] & key 4. colole is 1 and A[parent(i)] (A[i]) 5. do exchange A[i] & A[parent(i)] 6. is PARENT(i) The procedure HEAP-INICREASE—KEY complements the INICREASE—KEY operation. The auronay time of HEAP-INICREASE—KEY on plements the INICREASE—KEY operation. The auronay time of HEAP-INICREASE—KEY on plements from John of County from a fine of HEAP-INICREASE—KEY on an n-clement heap is O(1697) because the path traced from node is		1. return 1[1]
(a) HEAP_EXTRACT_MAX (A) 1. If heap-size[A] < 1 2. Then error" heap enologisted" 3. Then error" heap enologisted" 5. heap-size[A] < heap-size[A] 6. MAX_HEAPIFY (AXI) 7. return one The procedure HEAP_EXTRACT_MAX or prements the EXTRACT_MAX operation. The runner fine of HEAP_EXTRACT_MAX is O(1297), Decounce MAX_HEAPIFY tokes O(1297) The procedure HEAP_INCREASE-ICEY (A, i, key) 1. If key (A i, i) 2. then error" geod key is smaller than current key". 3. A i i = key 4. colole ix 1 and A [parent (i)] < A [i] 5. do exchange A [i] The procedure HEAP_INCREASE-KEY amplements the INCREASE KEY operation The runners time of HEAP-INCREASE KEY on plements the INCREASE KEY operation The runners time of HEAP-INCREASE KEY on on n-clement heap is O(1697) because the path traced from node i		The procedure HEAR MAXIMUM complements the maximum operates.
1. If heap-size[A] < 1 2. Ahan error" heap anderflow" 3. MAX HEAP Size[A] < heap-size[A] - 1 5. heap-size[A] < heap-size[A] - 1 6. MAX HEAP IFY (Axi) 7. Yeturg max The procedure HEAP EXTRACT MAX amplements the EXTRACT-MAX operation The running time of HEAP-EXTRACT MAX is O(Ityn). Decounce MAX-HEAPIFY takes O(Ioyn) The procedure HEAP INICREASE-ICEY (Axi, key) 1. If key (A [i] 2. then error" gets key is smaller than current key". 3. A [i] + ky 4. colore i > 1 and A [PARENT (i)] < A [i] 5. do exchange A [i] + A [PARENT (i)] 6. i PARENT (i) The procedure HEAP-INICREASE-KEY implements the INICREASE-KEY operation. The running time of HEAP-INICREASE KEY and an n-clevent heap is O(Ioyn) because the path traced from made i	5 -	en OCI) Heroge.
1. If heap-size[A] < 1 2. Ahan error" heap anderflow" 3. MAX HEAP Size[A] < heap-size[A] - 1 5. heap-size[A] < heap-size[A] - 1 6. MAX HEAP IFY (Axi) 7. Yeturg max The procedure HEAP EXTRACT MAX amplements the EXTRACT-MAX operation The running time of HEAP-EXTRACT MAX is O(Ityn). Decounce MAX-HEAPIFY takes O(Ioyn) The procedure HEAP INICREASE-ICEY (Axi, key) 1. If key (A [i] 2. then error" gets key is smaller than current key". 3. A [i] + ky 4. colore i > 1 and A [PARENT (i)] < A [i] 5. do exchange A [i] + A [PARENT (i)] 6. i PARENT (i) The procedure HEAP-INICREASE-KEY implements the INICREASE-KEY operation. The running time of HEAP-INICREASE KEY and an n-clevent heap is O(Ioyn) because the path traced from made i	(2)	HEAP- EXTRACT-MAX (A) 3211 AA3H - CAM SEARCH ANTI
A. other error heap emoler flow." S. onar L ACIT 4. ACIT Leap-size [AT] 5. heap-size [AT] Leap-size [AT] 5. heap-size [AT] Leap-size [AT] 6. MAX HEAPIFY (AXI) 7. Yeturn room. The procedure HEAP-EXTRACT MAX oraple rooms the EXTRACT MAX OPERATION. The running fime of HEAP-EXTRACT MAX is O(Irgn). Decounte MAX HEAPIFY touch O(Irgn) The procedure HEAP-INICREASE-ICEY (A, i, key) 1. If key (ACI) 2. then error of peop key is somether than current key". S. ACIT (Rey 4. collic in and AC PARENT (I) I (ACIT 5. do exchange ACIT ACPARENT (I) 6. in PARENT (I) The procedure HEAP-INICREASE-KEY oraple means the INICREASE-KEY Operation. The running time of HEAP-INICREASE-KEY and an n-element heap is O(Irgn) because the path traced from made i	- 11	
S. MAX. HEAPLEY CAYL) 5. Neap-SizeCAJ & heap-sizeCAJ 6. MAX. HEAPLEY CAYL) 7. Weturn man. 7. Weturn man. 7. The procedure HEAP-EXTRACT MAK omplements the EXTRACT-MAK operation. 7. The running time of HEAP-EXTRACT MAK is O(lign), Decount MAX. HEAPLEY tokes O(logn) 7. The procedure HEAP-INICREASE-ICEY (A, i, keg) 1. If keg < A[i] 2. then error in new key is smaller than current kegs. 3. A[i] + Rey 4. coh le ix1 cond A[PARENT(i)] < A[i] 5. do exchange A[i] + A[PARENT(i)] 6. ix PARENT(i) 7. The procedure HEAP-INICREASE-KEY on plements the INICREASE-KEY operation. 7. The running time of HEAP-INICREASE-KEY and on n-element heap is O(logn) because the path traced from made i	- 11	
4. ACIJ AP heap-size[A] 5. heap-size[A] h		
5. heap-size[A] < heap-size[A]-1 6. MAX_HEAPIFY (Axi) 7. Yeturg man. The procedure HEAP-EXTRACT MAX complements the EXTRACT-MAN operation. The running time of HEAP-EXTRACT MAX is O(light), Decades max-HEAPIFY tomas O(loght) 1. If key (A[i] 2. then error geo key is smaller than current key". 3. A[i] - Ry 4. culture is 1 and A[parent(i)] (A[i]) 5. do exchange A[i] 6. PARENT(i) The procedure HEAP-INCREASE-KEY amplements the INCREASE-KEY operation. The aurung time of HEAP-INCREASE-KEY and an n-clement heap is O(loght) because the path traced from made i		
The procedure HEAP_EXTRACT_MAK complements the EXTRACT_MAK operation. The running time of HEAP_EXTRACT_MAK is O(leg n). Deconcere max_HEAPIPY tokes O(log n) The procedure HEAP_IDICREASE_ICEY (A, i, key) 1. If key (Ati) 2. then error get key is smaller than current key". 3. Ati] < key 4. colore i>1 cood At parent(i)] < Ati] 5. do exchange A[i] <> At parent(i)] The procedure HEAP_INICREASE_KEY complements the INICREASE_KEY operation. The running time of HEAP_INICREASE_KEY on an n-clement heap is O(log n) because the path traced from made i		5. heap-size[A] - heap-size[A]-100
The procedure HEAP-EXTRACT MAX complements the EXTRACT-MAK operation The running time of HEAP-EXTRACT MAX is O(light). Decoucse MAX-HEAPIRY times O(loght) The procedure HEAP-INICREASE-ICEY (A, i, Key) 1. If key (ACi) 2. then error pero her is smaller than current key. 3. ACi] + key 4. coficie is 1 and AC PARENT (i)] < ACi] 5. do exchange ACi] AC PARENT (i) The procedure HEAP-INICREASE-KEY complements the INICREASE-KEY operation. The running time of HEAP-INICREASE-KEY on an n-clement heap is O(loght) because the path traced from nade i	756	3.3 MAX HEAPIPY CAND - STADADIN GALL SHOOD TO WAS
The procedure HEAP_EXTRACT_MAX complements the EXTRACT_MAX operation. The running time of HEAP_EXTRACT_MAX is O(109 n). Becomesse MAX_HEAPIFY times O(109 n) The procedure HEAP_IDICREASE_ICEY (A) is keep) 1. If key (AIi) 2. then error geto her is smaller than current keep. 4. cofile is 1 and AI PARENT(i)] (AII) 5. do excharge AIII (i) 6. it PARENT(i) The procedure HEAP_INICREASE_KEY omplements the INICREASE_KEY operation. The running time of HEAP_INICREASE_KEY on an n-clement heap is O(109 n) because the path traced from nade i		
The procedure HEAP-EXTRACT MAK complements the EXTRACT-MAK operation. The running time of HEAP-EXTRACT MAK is O(1000). Decoucce MAK-HEAPIRY tokes O(100 n) The procedure HEAP-IN/CREASE-KEY (A) is key) 1. If key < A[i] 2. then error geod key is somether than current key. 3. A[i] < key 4. cohole i>1 cond A[parent(i)] < A[i] 5. do eschange A[i] <> A[parent(i)] 6. ich PARENT(i) The procedure HEAP-IN/CREASE-KEY omplements the INCREASE-KEY operation. The running time of HEAP-IN/CREASE-KEY on an n-clement heap is O(100 n) because the poth traced form node i		
The running fime of HEAP-ENTRACT MAK is O(log n). Beconcere make HEAPIRY tokes O(log n) The procedure HEAP-INICREASE-ICEY (A, i, key) 1. If key (Ali) 2. then error upen key is somether than current key". 3. Alil key (Ali) 4. cohile in a cong Alenger (i) I (Ali) 5. do exchange Alil Aleparent (i) 6. in PARENT (i) The procedure HEAP-INICREASE-KEY complements the INICREASE-KEY operation. The running time of HEAP-INICREASE-KEY on an n-clement heap is O(log n) because the path traced from node i		
The running time of HEAP-ENTRACTMAX is O(1099). Becomese MAK-HEAPIRY takes O(1099) The provedence HEAP-IDICKEASE-KEY (A, i), Key) 1. If key < A[i] 2. Hen error peo key is somether than current key. 3. A[i] + key 4. coh, le i>1 cond A[PARENT(i)] < A[i] 5. do exchange A[i] + A[PARENT(i)] 6. ix PARENT(i) The provedence HEAP-INICKEASE-KEY complements the INICKEASE-KEY operation. The running time of HEAP-INICKEASE KEY on an n-element heap is O(1099) because the path traced from node i		EXTRACT-MAR operation South to free to
The provider HEAP-IN/CREASE-KEY (A, ii, key) 1. If key (A[i] 2. then error of key is smaller than current key". 3. A[i] (key) 4. cohile i>1 cend A[parent(i)] (A[i]) 5. do exchange A[i] (A[parent(i)]) 6. it PARENT(i) The procedere HEAP-IN/CREASE-KEY om plements the IN/CREASE-KEY operation. The running time of HEAP-IN/CREASE-KEY on an n-element heap is O(log n) because the path traced form nade i		The running time of HEAP-EXTRACT-MAX is O(1097). Becourse
The provider HEAP-INCREASE-KEY arm plements the INCREASE-KEY approach to a content to the conten		MAX-HEAPIFY takes O(1097)
The procedure HEAP-IDICREASE-ICEY (A, i, key) 1. If key (A[i] 2. then error open key is smaller than current key". 3. A[i] ← key 4. coh, le i > 1 cond A[parent(i)] < A[i] 5. do enchange A[i] ↔ A[parent(i)] 6. i ← PARENT(i) The procedure HEAP-INICREASE-KEY omplements the INICREASE-KEY openation. The running time of HEAP-INICREASE-KEY on an n-element heap is O(log n) because the path traced from nade i		
1. If key (A[i] 2. then error is now key is smaller than current key". 3. A[i] + key 4. while i>1 and A[parent(i)] < A[i]. 5. do enchange A[i] + A[parent(i)] 6. i) + PARENT(i) The procedure HEAP-INICREASE-KEY complements the INICREASE-KEY operation. The running time of HEAP-INICREASE-KEY on an n-clement heap is O(log n) because the path traced from node i	7	
1. If key (A[i] 2. then error is now key is smaller than current key". 3. A[i] + key 4. while i>1 and A[parent(i)] < A[i]. 5. do enchange A[i] + A[parent(i)] 6. i) + PARENT(i) The procedure HEAP-INICREASE-KEY complements the INICREASE-KEY operation. The running time of HEAP-INICREASE-KEY on an n-clement heap is O(log n) because the path traced from node i		HEAP-INICREASE-KEY (A, i), Key)
2. then error open key is somether than current key". 3. A [i] < key 3 4. cohole i>1 cond A [PARENT (i)] < A [i] 5. do exchange A [i] <> A [PARENT (i)] 6. i) <- PARENT (i) The procedeure HEAP-INCREASE-KEY complements the INCREASE-KEY operation. The running time of HEAP-INCREASE-KEY on an n-element heap is O(log n) because the path traced from nade i	- 11	
9. colore in a cond Alparent (i)] < Alignon (ii) 5. do enchange Alight Alparent (ii) 6. it PARENT (ii) The provedence HEAP-INICREASE-KEY complements the INICREASE-KEY operation. The running time of HEAP-INICREASE-KEY on an n-element heap is Ollogn) because the path traced from node i	2	
4. consile is 1 cond Alphanent (i)] < Alijo 5. do exchange Alijo Alparent (i)] 6. in Parent (i) The proceder HEAP-INCREASE-KEY omplements the INCREASE-KEY operation. The running time of HEAP-INCREASE-KEY on an n-element heap is Octogn) because the path traced from node i	8	. ATiTE key (2)
5. do enchange A[i] (>) A[PARENT(i)] 6. it PARENT(i) The protectione HEAP-INICREASE-KEY complements the INICREASE-KEY operation. The running time of HEAP-INICREASE-KEY on an n-clement heap is O(log n) because the path traced from node i	- 11	
The procederre HEAP-INICREASE-KEY complements the INICREASE-KEY operation. The running time of HEAP-INICREASE-KEY on an n-element heap is Octogn) because the post traced from node i	5	
The procederal HEAP-INICREASE-KEY complements the INICREASE-KEY operation. The running time of HEAP-INICREASE-KEY on an n-element heap is Octogn) because the path traced from node is	6	
The running time of HEAP-IN/CREASE-NEW on an n-element heap is Octogn) because the path traced from node i		
The running time of HEAP-IN/CREASE-NEW on an n-element heap is Octogn) because the path traced from node i	7	be procedure HEAD INCREMENT LESS
The running time of HEAP-IN/CREASE-NEW on an n-element heap is Octogn) because the path traced from node i		
heap is Octogn) because the path traced from node i	- 11	
	- / ·	te running time of HEAP-/N/CREASE-KEY on an n-climent
10 800 t hos 100 gth 0(109 7)	11	
The state of the s	40	goot how langth o(1597)
II was a second of the second	-	a_{i}

