Неар BWI, 10 KIN, ZO JNB, IS MNL, ZZ SEA, 30 DEL, 40 AKL, 25 LIM, 35 CAT, 24 MEX, 36 Heap: a binary free such that order the value in a node is E the value in any children (min-heap) shape all levels but possibly the last are full and nodes are as far left as possible

Heap BWI, 100 BL KIN, 20 MNL, 22 35-JNB,15 2 SEA,30 DEL,40 AKL,25 LIM, 35- CAT, 24 MEX, 36 Heap: a binney free such that order the value in a node is E the value in any children (min-heap) shape all levels but possibly the last are full and nodes are as far left as possible

Heap BWI, 100 LEFT(i)=2:+1 KIN, 20 JNB, 15 Z MNL, 22 SEA, 30 DEL, 40 AKL, 25 RIGHT(i) = Zi + ZLIM, 35- CAE, 24 MEX, 36 DEL AKL LIM BUI KIN JNB MNL (AL 120 22 40 25 15 35 30 24 Heap: a binney free such that order the value in a node is & the value in any children shape all levels but possibly the last are full and nodes are as far left as possible SEA, 30° CAI, 24° JNB, 15' SEA, 30² BWI, 10¹ KIN, 20⁴ AKL, 25⁵ MEX, 36 CAT , 24 MEX, 36 JNB, 15 AKL, 25 LIM, 35 DEL, 40 MAL,22 LIM,35 DEL, 40 BWI, 10 KW, 20 MML, 22

AKL 6 Heap BWI O DEL 5 SEA Y LIM 7 MNL 3 BWI, 100 LEFT(i)=Zi+1 KIN, 20 MNL, 22 35-2 RIGHT (;) = Z; +Z JNB,15 2 CAL 8 MEX 9 MNL, 22 SEA, 30 DEL, 40 AKL, 25 LIM, 35 CAT, 24 MEX, 369 PARENT(;)=[=] KIN JNB 2 6 7 DEL AKL LIM BUT KIN JUB MAL SEA CAL MEX 40 25 35 22 30 24 Henp: a binney free such that order the value in a node is \leq the value in any children shape all levels but possibly the last are full and nodes are as far left as possible CAI, ZY SEA, 30[°] JNB, 15' SEA, 30² BWI, 10 MML, 22 AKL, 25⁵ MEX, 36 CATL ,24 MEX. 36 JNB, 15 AKL, 25 LIM, 35 DEL, 40 LIM, 35 DEL, 40 BWT, 10 KW, 20 MAL, 22 KIN, 20

Неар 34 24 engueve (key, pri): i < n add (key, pri) at location (i) while priority at i < priority at PARENT(i) Swap i, PARENT(i)

Неар 10 ZD 15 22 25 40 34 24 30 i < n add (key,pri) at location (i) while priority at i < priority at PARENT(i) swap i, PARENT(i) i < PARENT(i) enqueve (key, pri):

Неар 10 20 15 zź 40 25 34 24 30 36 engueve (key, pri): add (key, pri) at leastion (i) while sump i, PARENT(i) i < PARENT(i)

Heap 10 15 22 ` 25 34 40 24 36 30 engreve (key, pri): i < n worst case $\Theta(\log n)$ add (key, pri) at location (i) while i>0 and priority at i < priority at PARENT(i) swap i, PARENT(i) < I iteration por level i < PARENT(i) $\Theta(\log n)$ levels $n \in n + 1$ O(1) per itembion

Heap 20 22 34 24 36 15 40 25 dequeue (): remember item at root worst case $\Theta(\log n)$ move last node to root remember item at root ntn-l : -0 while LEFT(i) < n and priority at i > priority at one of its children-swap i with smallest child O(1) per iteration i < former index of smallest child worst case Ollog n iterations) change_priority: find index i of item to change use loop from dequeve if priority value T, loop from engueve if J

Heap MAL, ZZ KIN 1 Z JNB 6 AKL 98757 MEX 3~8,152 CAI MAL SEA, 30 DEL 40 ANIZE DEL LIM 41M, 34 CAE, 24 MEX, 36 40 SEA LEX change-priority (MNL, 5) de queve (): remember item at root worst case $\Theta(\log n)$ move last node to root remember item at root ntn-l $i \leftarrow 0$ while LEFT(i) < n and priority at i > priority at one of its children-swap i with smallest child O(1) per iteration i < former index of smallest child worst case Ollog n iterations) change_privrity: find index i of item to change worst case O(log n) use loop from dequeve if priority value 1, loop from engueve if J

Dijkstra's Algorithm for each v color[v], pred[v], d[v] \leftarrow IN_QUEUE, NIL, ∞ 10 d[s] ← 0 Q ← new PriorityQueue(d) 0 while Q not empty $u \leftarrow dequeue (Q)$ for each outneighbor v of u if $color[v] = IN_QUEUE$ and d[v] > d[u] + w(u, v)change_priority(Q, v, d[u] + w(u, v)) $d[v] \leftarrow d[u] + w(u, v)$ $pred[v] \leftarrow u$ $\texttt{color[u]} \leftarrow \texttt{DONE}$ vertex B X Z 3 priority(k) O 063 001 00 By prikessor D -4 5 00 11 00 7 2