in the same A-3. ι. 2 (B 0 Decision Tree. . AB CA AC 5 3 ABC ABD ACE ACD DDC BOA 50 AB CP ABDC A CBD AD(BC) ACDB DOCB 31 X. HI ABCDA ABDCA ACBDA ACOBA ADBCA ADCBA Rough Algorithm lef (at to go (city, history 13): if housey [wild not in history history (city) = cost to go to A 20 D B D A 0 2 1 **1** => Cost of visiting city in - City B 2 0 B 2 (two ates). Ó 1 C 0 8 d 3P BI 2 0

<u>e</u>

6

From the code. AB (2) 4C C13 (1) an ABC ABD ACB ACD ADB ADC (2+2) CKD (Ital) (3) (3) (3) ABCD ABDC ACBD ACOB ADBC ADC B (5) (6) (3+xx) (K+d) (3+X) (3+06 ABCDA ADCE ARDCA ACBDA ACDBA ADBCA (4+x) (7) 5+2 (5+a) (7) (h+x) if a ~[1,2] The paths ACBDA & ADBCA will be optimal since the cost-to-go for both the paths will be 6. a) i) recover Path. input: 5, 9, pred. SEV, gev Pred is a dictionary. output: to sequence of vertices on the optimal po tel it be a list of seclement ev.