```python
# X_2007_bzip.py

def _1_occurrences(t):
    r = 256*[0]
    for x in t:
        r[x] += 1
    return r

def _2_min(t):
    r = _1_occurrences(t)
    entier_min = 0
    for k in range(1, 256):
        if r[k] < r[entier_min]:
            entier_min = k
    return entier_min

def _3_tailleCodage(t):
    n = len(t)
    taille = n + 1
    if t[0] == t[1]:
        taille += 1
    for j in range(2, n):
        if t[j] == t[j-1]:
            if t[j] == t[j-2]:
                taille -= 1
            else:
                taille +=1
    return taille

def _4_codage(t):
    marqueur = _2_min(t)
    code, compteur = [marqueur], 1
    t.append(-1)                    # sentinelle
    for j in range(1, len(t)):
        if t[j] == t[j-1]:
            compteur += 1
        else:
            if compteur == 1:
                code.append(t[j-1])
            else:
                code = code + [marqueur, compteur - 1, t[j-1]]
                compteur = 1
    t.pop()                         # sentinelle out
    return code

def _4_codage_bis(t):
    marqueur = _2_min(t)
    code, compteur = [marqueur], 1
    for j in range(1, 1 + len(t)):
        if j < len(t) and t[j] == t[j-1]:
            compteur += 1
        else:
            if compteur == 1:
                code.append(t[j-1])
            else:
                code = code + [marqueur, compteur - 1, t[j-1]]
```

```
055|                 compteur = 1
056|     return code
057|
058| def _4_decodage(t1):
059|     marqueur = t1[0]
060|     t = []
061|     k = 1
062|     while k < len(t1):
063|         if t1[k] != marqueur:
064|             t.append(t1[k])
065|             k += 1
066|         else:
067|             repetitions = t1[k+1] + 1
068|             t = t + repetitions*[t1[k+2]]
069|             k += 3
070|     return t
071|
072| def _5_compRotations(t, i, j):
073|     n = len(t)
074|     for k in range(n):
075|         if t[(i+k) % n] > t[(j+k) % n]:
076|             return 1
077|         if t[(i+k) % n] < t[(j+k) % n]:
078|             return -1
079|     return 0
080|
081| def _5_triRotations(t):
082|     n = len(t)
083|     r = [k for k in range(n)]
084|     for i in range(1, n):
085|         a, j = r[i], i-1
086|         while j >= 0 and _5_compRotations(t, a, r[j]) == -1:
087|             r[j+1] = r[j]
088|             j -= 1
089|         r[j+1] = a
090|     return r
091|
092| def _6_codageBW(t):
093|     n = len(t)
094|     r = _5_triRotations(t)
095|     codeBW = [t[r[k] - 1] for k in range(n)]
096|     for k in range(n):
097|         if r[k] == 1:
098|             clef = k
099|     codeBW.append(clef)
100|     return codeBW
101|
102| # Q7  O(n**2 * ln(n))
103|
104| def _9_triCarsDe(t):
105|     occ = _1_occurrences(t)
106|     triCars =[]
107|     for i in range(256):
108|         for j in range(occ[i]):
109|             triCars.append(i)
110|     return triCars
```

```
111|
112| def _10_trouverIndices_lin(t):
113|     n = len(t)
114|     occ = _1_occurrences(t)
115|     triCars = _9_triCarsDe(t)
116|     cumul = 256*[0]
117|     for j in range(1, 256):
118|         cumul[j] = cumul[j-1] + occ[j-1]
119|     indices = n*[0]
120|     for i in range(n):
121|         c = cumul[t[i]]
122|         indices[c] = i
123|         cumul[t[i]] += 1
124|     return indices
125| # complexité linéaire
126|
127| def _10_trouverIndices(t1):
128|     n = len(t1)
129|     t2 = t1[:]
130|     triCars = _9_triCarsDe(t2)
131|     indices = []
132|     for i in range(n):
133|         k = 0
134|         while t2[k] != triCars[i]:
135|             k += 1
136|         indices.append(k)
137|         t2[k] = -1
138|     return indices
139| # complexité quadratique
140|
141| def _11_decodageBW(t1):
142|     clef = t1.pop()
143|     indices = _10_trouverIndices(t1)
144|     t = []
145|     position = clef
146|     for _ in range(len(t1)):
147|         t.append(t1[position])
148|         position = indices[position]
149|     return t
150|
151|
152| test = [0,0,3,2,3,3,3,3,3,3,5]
153| code = _4_codage(test)
154| code_bis = _4_codage_bis(test)
155| decode = _4_decodage(code)
156| print(test == decode)
157| print(code == code_bis)
158| texte_brut_1 = 'concours'
159| texte_brut_2 = 'concours_de_l_ecole_polytechnique'
160| texte_1 = [ord(c) for c in texte_brut_1]
161| texte_2 = [ord(c) for c in texte_brut_2]
162| print(_10_trouverIndices(texte_1))
163| print(_10_trouverIndices_lin(texte_1))
164|
165| cd_2 = _6_codageBW(texte_2)
166| dec_2  =_11_decodageBW(cd_2)
```